FiBL

SOLm Model Documentation

Adrian Muller, Anita Frehner, Catherine Pfeiffer, Simon Moakes, Christian Schader

22.11.2020

Contact: adrian.mueller@fibl.org

Research Institute of Organic Agriculture FiBL | Ackerstrasse 113 | Postfach 219
5070 Frick | Switzerland | Phone +41 62 865 72 72 | info.suisse@fibl.org | www.fibl.org

Table of Contents

Inhaltsverzeichnis

Acknowledgment: A first version of this documentation had been written in February 2020 as part of the deliverable D4.1 of the research project "UNISECO", funded by European Union's Horizon 2020 research and innovation programme under grant agreement N° 773901.

1. Introduction

SOLm is a mass- and nutrient-flow model capturing the global food system on the level of geographic units (default: countries), linking production, consumption and trade, with the aim to derive and analyse the food system's input use, outputs and sustainability impacts for a wide range of future or counterfactual current food system scenarios. This section describes SOLm in general terms. Further details are given in the subsequent sections 2 and 3.

1.1 History

SOLm has originally been developed in the context of a project on sustainable and organic livestock production for the FAO, running from 2011-2013. This coined the name of the model, i.e. "**S**ustainability and **O**rganic **L**ivestock **m**odel – **SOLm**"[1]. It has subsequently been applied and developed further in a number of projects with a focus on food waste (still with the FAO), the alpine region ("Projekt Alpenraum", using refined data for Switzerland and Austria) and Switzerland (a Swiss National Science Foundation project on "Sustainable and Healthy Diets", focusing on refined Swiss consumption data and adding health aspects to the model), and in the EU-Horizon2020 project "UNISECO", with a focus on sub-country level analysis of agro-ecological production systems for the EU. It is currently available in the thoroughly reworked sixth version from autumn 2019, which has continuously been improved since then. From now on, the acronym "SOLm" or the term "the SOL model" is used (in German: "SOLm" or "das SOL-Modell") and others, such as SOL-m, Sol-m, SOL-M, Sol-M, "the SOL-model", "the SOLm-model" or such are to be avoided. If the version of the model is important to be made explicit, the indication "V#" is added to "SOLm", resulting in names such as "SOLmV6" (the current version) or "SOLmV2" (the version used for the publications (Schader, Muller et al. 2015, Muller, Schader et al. 2017), for example).

The first version of this documentation has been written in the EU-Horizon2020 project "UNISECO" and is part of the project deliverable D4.1.

1.2 Access to and working with the model

Currently, the main version of the model is stored, back-upped and runs on a terminal server (fisrvts02[2]) at the Research Institute of Organic Agriculture FiBL, which can be accessed by employees of FiBL via a vpn-and remote-desktop connection. For people not working at FiBL, it is possible to get local copies via BitBucket (for the code and the documentation: non-public repository "SOLmV6": https://bitbucket.org/FiBL-Socioeconomics/solmv6) and an ftp-server (for the data: ftp://austausch1.fibl.ch/DataFiles_SOLmV6/, password protected). The master copy of the model is stored on the terminal server at FiBL and the files on BitBucket as well as on the ftp-server are regularly updated with the newest code and data files from the terminal server. On the ftp-Server, there is also a sub-folder DataForConsistencyChecks, which contains files that are used for consistency checks of SOLm.

Using "git" (https://git-scm.com/downloads) and Sourcetree (https://www.sourcetreeapp.com/) on your computer, you can participate in using the files from BitBucket with version control and any changes appearing on the BitBucket-repository will be indicated within Sourcetree on your local

[1] In 2019, SOLm has unofficially been identified to signify "**S**ooner **O**r **L**ater, maybe", with corresponding consequences on how the model is called…

[2] Unofficially named "ROSE – **R**oom for **o**utputs from **s**tubborn **e**ndurance"

computer, while you can also upload changed files to the BitBucket repository, if you have writing-rights.

If you are interested in getting access to and working with SOLm, please contact Adrian Muller (adrian.mueller@fibl.org). We will then provide the details for accessing the data on the ftp-server and invite you to the Bitbucket-repository, pending some use and collaboration agreements to be set up on an individual basis. When running the model for the first time, some settings need to be changed to read all data: in ___V6_SteeringFile1_ModelInitialisation.gms, all settings in section 2.1, that are on “NO” per default need to be set to “YES” when running it for the first time, to read all data that is provided in xlsx- and csv-files. After this, all these assignments need to be set back to “NO” again (as otherwise, the model takes too long when you run ___V6_SteeringFile1_ModelInitialisation.gms again, e.g. to read some new specific data) and the data is then available in gdx-form that can be readily used.

1.3 General Structure

SOLm is a mass- and nutrient-flow model of the food system. It thus traces all mass and nutrient flows through the food system, from the inputs to agricultural production to the emissions from food waste disposal. Generally, the system analysed is understood as being in an equilibrium/static situation for several consecutive years. Thus, crop rotations, for example, are captured by allocating corresponding shares of the crops per hectare (cf. section 3.8) and animal herds are understood to be in a steady state of animals leaving the system and being replaced constantly. The data on animal numbers, hectares cropped, production quantities, etc. can thus be seen as the annual values observed on average over the period of several years. Thus, living animals is the numbers of animals that can be observed to be living at any point of time, while producing animals are those that produce within a year (thus, if there is a 2.5-3 turnover rate in pig production per year, producing animals count all of those, i.e. the number of producing animals can be much higher than the number of living animals – e.g. for meat production, and for chicken in particular).

The basic structure of SOLm is captured in the following two figures Figure 1 and Figure 2. Starting from a certain area for agricultural production cropped with a number of crops or managed as grasslands, the corresponding outputs are derived, comprising main outputs such as grains but also by-products such as residues. These outputs are then exported from agricultural production to be used as food, as feed for animal production, otherwise (e.g. for fiber or bioenergy), or they are recycled to the agricultural areas (e.g. as residues left on the field). Together with other inputs (e.g. mineral fertilizers, pesticides) and all emissions and impacts from the production activities, all this captures the land use/plant production part of the model. Animal production is captured similarly, tracing the flows from feed from croplands and grasslands to outputs from animal production systems. Part of the outputs are recycled to animal production (e.g. whey from cheese production fed to pigs), part is used in crop production (e.g. manure), and part is exported from agricultural production to be used as food or otherwise (e.g. wool). Together with other inputs (e.g. antibiotics, drinking water for animals) and all emissions and impacts from the animal production activities, all this captures the animal production part of the model. The nutrient flows in feed and manure link animal and crop production and all these parts together describe the production part of SOLm.

This combined land use/plant-animal production system is located within geographic boundaries – the default is country level, but provided adequate data is available, it can be refined to arbitrary sub-country levels. The results from the calculations can then be aggregated to country groups, world regions or a global picture.

Forest areas and forestry can be captured in the same structure as land use/plant production, i.e. as specific activities on land areas; fish and seafood production are captured in the same structure as animal production, i.e. as animal production activities utilizing feed and other inputs to produce some outputs. It is also possible to add new activities, such as production of artificial meat or vegetables in soil-less vertical farms, as such can also be captured by tracing the inputs required, the outputs produced and the emissions and impacts incurred.

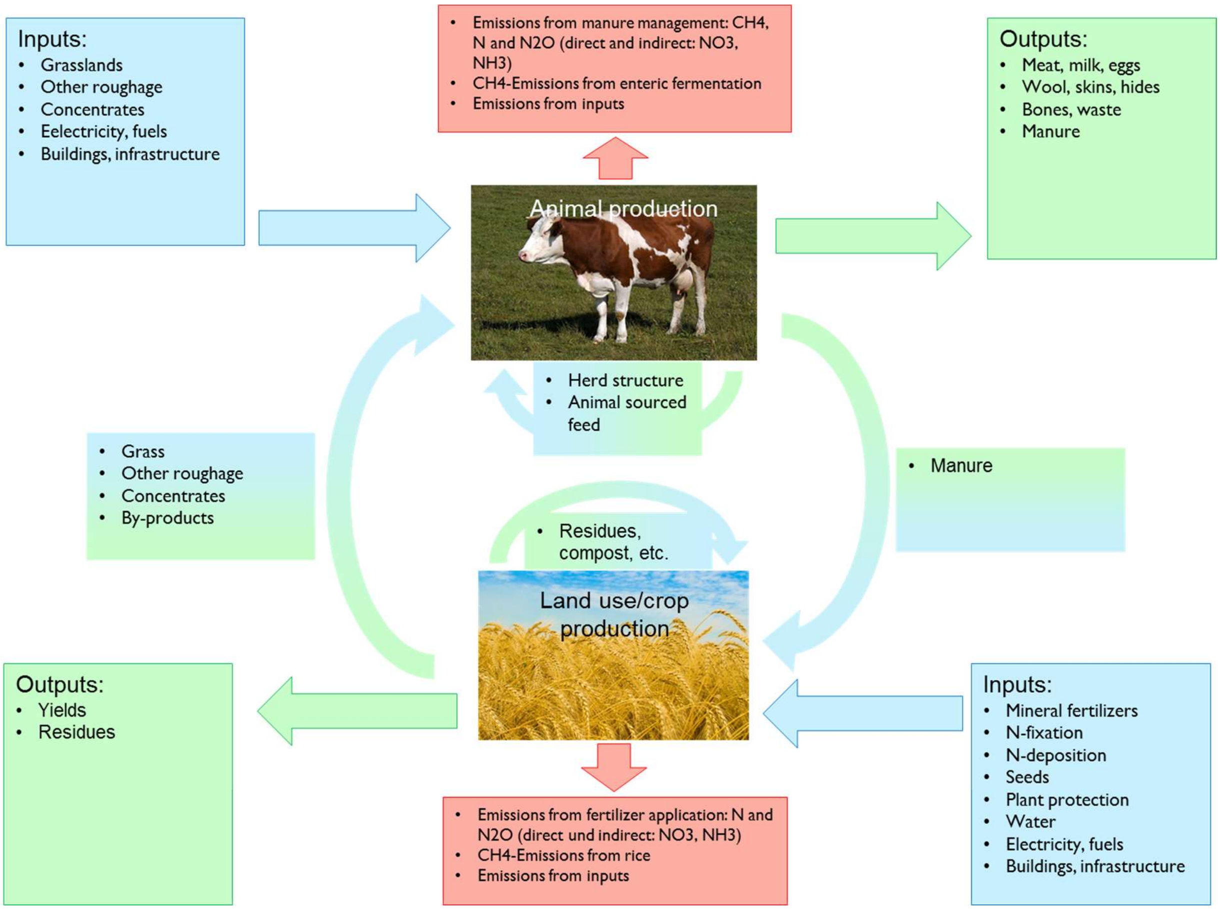

Figure 1: Structure of the agricultural production in SOLm

The calculations thus capture the agricultural areas and the numbers of animals and the related inputs, emissions, impacts and outputs in each geographic region of interest. Inputs can thereby also be imported from outside the systems, and impacts can also become relevant outside the system. This is then better captured by adopting the food-system view on the SOLm, model, schematically displayed in Figure 2, which is complementary to the production-focus just presented above. In the food-system view, the mass and nutrients flows are traced starting from the “domestically available quantity” of commodities. This quantity stems from domestic plant or animal production (depending on the commodity) and from imports and stock changes, while exports or allocation to building up stocks reduce the domestically available quantity. This domestically available quantity is then used for various utilizations such as “food”, “feed” or also “waste” or “bioenergy” (here in Figure 2 captured via “other”). The share used for food determines the food availability (which food, as said, stems not only from domestic production within this region but also from imports or stock changes), captured by physical quantities but also by macro- and micro-nutrient supply, such as calories, proteins, vitamins, etc. The share used for feed determines how much feed is available for animal production within each geographic region (which feed, again, stems not only from domestic production within this region but also from imports or stock changes). Finally, as already indicated in the production view above, both plant and animal production result in a number of emissions and impacts, which – in the food systems view – are seen as emissions and impacts from (different parts of) the food system.

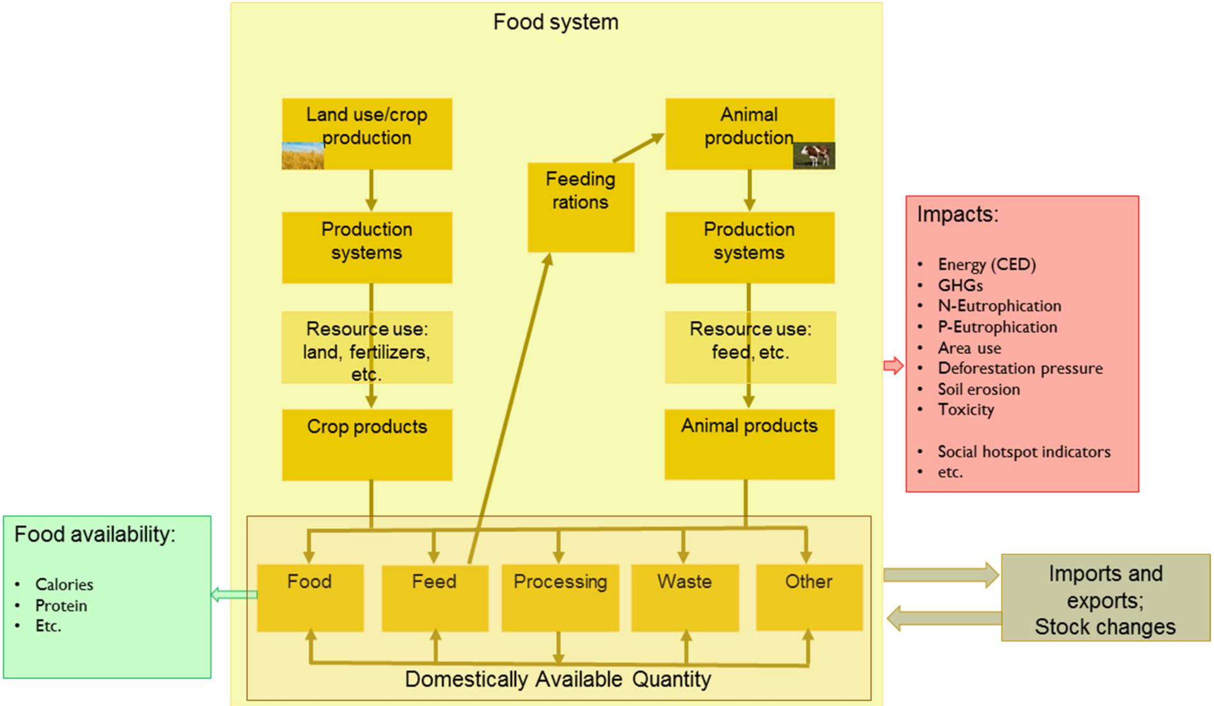

Figure 2: Structure of the food system in SOLm ("CED" is Cumulative Energy Demand, an energy use indicator from Life Cycle Analysis (LCA); "GHG" is Greenhouse Gas; "N" is Nitrogen; "P" is Phosphorus)

There is a number of entities used in the code in SOLm. The core entities used and shortly presented here are **activities**, **commodities** and **regions** (formally, these are "sets" in the programming language GAMS, cf. section 2.1). **"Activities"** are as all actions, technologies, transformations that produce a number of outputs with a number of inputs, thereby also causing emissions and other sustainability impacts. In SOLm we differentiate further between **"plant activities"** and **"animal activities"**. "Plant activities" cover crop and grassland production, where one key input is land area and where the basic metric to measure the size of an activity is the land area covered by the activity. Relative values, such as input factors, emission factors, etc. are then also primarily given on a per area base, but can easily be linked to outputs via yields. "Animal activities" cover all livestock operations, i.e. all production settings where by means of some animals some feed and other inputs are transformed into a number of outputs. The basic metric for the size of an activity is the number of animal heads and many indicators are then also reported on a per animal head basis. Animal activities usually show the additional complexity of a "herd structure", which captures the number of different types of animals (different age groups, male/female animals, replacement animals, etc.) that have to be present to support a given number of producing animals. These animals do not produce anything but also need feed and cause emissions and impacts which need to be accounted for. Thus, a second view on animal activities is by building "animal production units", which combine one producing animal (e.g. a dairy cow) with a consistent proportion from all the other types of animals such as to allow for a stable population of producing animals over time (thus, per dairy cow, there is a certain share of replacement animals needed, a certain amount of calves are present, a certain share of sires for reproduction, etc.). Thus, a second view on animal activities is given by these animal production units (APU), where the basic metric for the size of an activity is then given by the number of APUs and key parameters are reported on a per-APU basis.

SOLm allows for a number of other activities besides plant and animal activities. In SOLmV6, these are Fish and Seafood activities, forest activities and other activities, each able to account for the specific aspects of these production activities. “Other activities” are coded very generally and not yet provided with data, but they would allow to cover activities such as insects, algae or cultured meat as soon as they would become the focus of interest in some model runs.

“Commodities” are the physical primary outputs or products from activities, including main and by-products (e.g. “Wheat grains” and “Straw” from the activity “Wheat”), as well as products derived from the primary products by one or several processing steps (e.g. “Wheat flour”, “Wheat bran”, “Starch from wheat”, etc.). The relation to primary products that are the direct outputs from activities is governed via extraction rates (e.g. 0.75 for wheat flour, when 0.750 tons of wheat flour may be derived from 1 ton of wheat grains, etc.) and the commodity trees linking the various commodities among each other by indicating which commodities are derived from which ones (e.g. “maize grains” are the primary product of the activity “maize”, “maize germs” are derived from “maize grains”, “maize germ oil” is derived from “maize germs” on a further level, etc.).

“Regions” refer to the geographical boundaries for which the data is provided. The default regions are countries (as used in FAOSTAT, for example), but they also cover sub-country level geographic areas, such as NUTS-2 regions, or counties, etc. For displaying results and certain analyses, aggregate regions are used, such as sub-continents, world regions or also a global picture aggregating all regions.

1.4 What does SOLm deliver and what not

Generally, SOLm delivers a number of indicators on areas and animal numbers, production levels, input use, emissions and other impacts of a wide range of agricultural production and food-system situations.

1.4.1 Option space, viability and scenarios

Examples of such situations are the conversion of a significant share or 100% of production to organic agriculture, the shift towards zero food-competing feed use (i.e. abandoning concentrate and forage feed specifically cropped on cropland, where food crops could be grown directly), a shift towards increased shares of fish and seafood in animal protein supply or a shift towards a healthier composition of food baskets. Often such situations involve quite drastic changes in production and consumption and are thus understood as investigations into food systems in the farther future, say in 2050 or 2100, then also using reference scenarios for the same time-scales, such as FAO or IPCC projections for 2050 (e.g. regarding population numbers, economic development, etc.). However, counterfactual analysis of the current food systems are also possible, e.g. by investigating a situation of 100% conversion to organic agriculture of the current Swiss or other national food systems, thus using the current situation as a baseline for comparison. In fact, due to historical data from FAO being available in the model (cf. section 3.1), it is in principle also possible to investigate counterfactual situations for past time periods. Generally, the term “baseline” is used to refer to one year or to a several-year time-period representing the “current” agriculture and food system. Due to data availability, this lies often some years in the past. The term “reference scenario” is used for a hypothetical, often future, situation, which can be considerably different from the situation today, e.g. a projection for the food system in 2050 as provided by the FAO. In the model runs, further hypothetical situations of the food system are then calculated and compared to the baseline or the reference scenario, depending on the context the calculations refer to (e.g. a counterfactual to a current situation or an alternative for the 2050 business as usual food system).

The results from the model runs can be framed by the notion of the "**option space**". Assume that we investigate, for example (as done in (Muller, Schader et al. 2017)), the consequences of a conversion to increasing shares of organic agriculture, combined with different levels of food-competing feed reduction and different levels of food waste reduction, all assessed in a context of increasingly strong climate change impacts on yields. The reference scenario for this example is the FAO business as usual projection for 2050 from 2012, as described in (Alexandratos and Bruinsma 2012). The results of all these changes can be framed by the various combinations of different parameter values along these four dimensions (each one combination providing one "option" for the future food system) as displayed in Figure 3, employing the example of "percentage change in cropland use with respect to the reference situation" as an impact indicator (dimension 1: shares of organic production of 0,20,40,60,80,100%; dimension 2: food-competing feed reduction by 0,50,100%; dimension 3: food waste reduction by 0,25,50%; dimension 4: climate change impact on yields resulting in yield increases till 2050 being reduced by 0,50,100% with respect to the reference forecasts). Figure 3 thus displays the "option space" for global cropland use of the combinations of changes along these four dimensions which can then be assessed to identify promising options (with strongly reduced land use) and rather negative options (with strongly increased land use).

In short, ***the option space is the totality of all options that emerge from combining different levels along a number of different key dimensions describing a food system of interest that are varied to explore the potential future or counterfactual to a baseline or reference scenario of this food system*** (in the example above: share of organic production, food-competing feed reduction, wastage reduction and climate change impact on yields, and all this is assessed for the global food system).

It has to be emphasized that the aggregation to global levels as done in this example potentially hides important regional patterns, as situations could arise, where additional local land use would be prohibitively high albeit total land use would decrease globally. SOLm however provides the data needed to also assess this, as it basically works on country (or – depending on data availability – even finer) level.

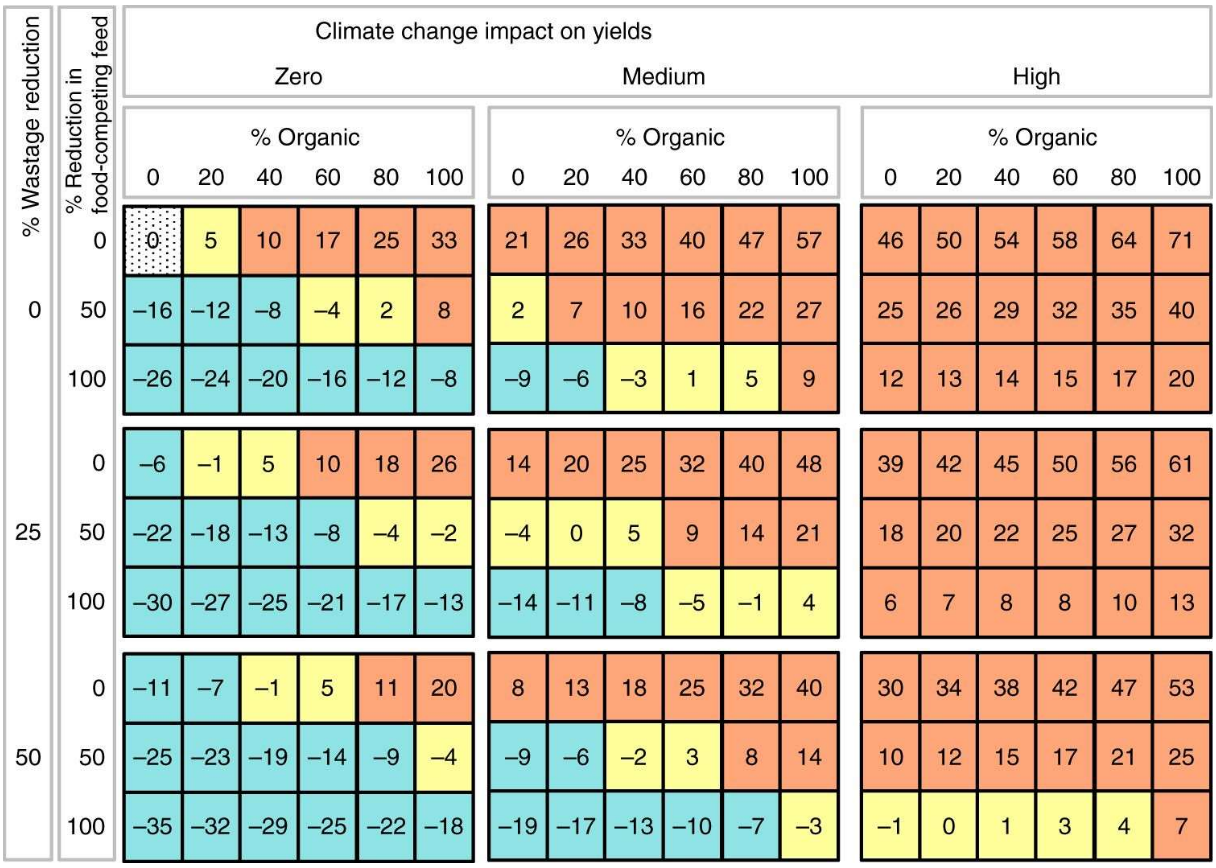

Figure 3: Example of an option space as the result from SOLm model runs (Source: (Muller, Schader et al. 2017) Original Figure Caption: Cropland area change. Percentage change in cropland areas with respect to the reference scenario. Scenarios differ in: organic shares (0–100%), impacts of climate change on yields (low, medium, high), food-competing feed reductions (0, 50, 100% reduced from the levels in the reference scenario), and wastage reduction (0, 25, 50% compared to the reference scenario). Colour code for comparison to the reference scenario value (i.e. 0% organic agriculture, no changes in livestock feed and food waste, dotted grey): > +5%: red, < −5% blue, between −5% and +5% yellow; in the reference scenario, cropland areas are 6% higher than in the baseline today.).

The option space can be provided for any indicator of interest, such as, besides “cropland use”, greenhouse gas emissions, nitrogen surplus, deforestation pressure, aggregate eco-toxicity, erosion risk, etc. When displaying an option space for a number of parameter combinations for one indicator, the notion of “viability” of an option (i.e. of a combination of certain parameter values) is also helpful in analysis and communication. “**Viability of an option**” refers to the performance of an option, i.e. the corresponding indicator value, in relation to some general external restrictions, that can be biophysically mandatory (e.g. global potential cropland area is physically constrained and any option resulting in a larger global land use than this upper limit is biophysically impossible, i.e. unviable) or (politically highly) desirable (e.g. related to planetary boundaries on greenhouse gas emissions, deforestation or nitrogen surplus for a safe operation space of humankind, and any option that would e.g. conflict with some deforestation reduction targets or with limiting global warming to 1.5 degrees would thus also have to be termed “unviable”. This judgment would however in this case not be based on biophysical terms but in relation to the goal to not transgress these boundaries for a safe operation space for humankind).

Finally, a third term is often used in the context of option spaces and viability of options, namely “**scenarios**”. In this context, we use scenarios to describe certain options or groups of closely similar options which we amend by a detailed description of which socio-economic and cultural, etc. dynamics may drive them, e.g. regarding the context of policies, peoples' preferences and values, economic development etc. without explicitly including this additional information in the model. ***Scenarios are thus telling the rich societal, economic, psychological and value-related (i.e. ethical) stories or narratives behind a special option or behind an sub-set of similar options in the option space.***

1.4.2 Decision structure and principle of "ceteris paribus – keeping everything else equal"

SOLm does not have an internal decision structure for decision making units (such as e.g. country-wise or global utility maximization in economic models or greenhouse gas minimization as an environmental goal, etc.). It is thus driven by the explicit exogenous assumptions set by the researchers running the model. As the scenario specification in the model is organized in its current version (for details, see section 3.4), these assumptions have to be quantified by the amount of area cropped by each crop and grassland type of relevance in each geographic region. Given this input, and a number of other assumptions that are provided as defaults or are also to be chosen by the researchers in the scenario specification, the model runs to derive plant production, feed use and related animal production, total food availability, waste and other utilization (e.g. bioenergy) and all related emissions and other impacts.

Given the absence of an internal decision structure for decision making units, the general philosophy in SOLm is to adopt a "ceteris paribus" approach, i.e. an approach that "keeps all else equal - or as close as possible" (to some baseline or reference situation) besides some core changes of central interest for a specific scenario. That is, besides the "core changes of central interest for a specific scenario" the approach is to keep this "number of other assumptions that are provided as defaults or are also to be chosen by the researchers in the scenario specification" mentioned above as close as possible to some baseline or reference situation, unless the research question to be addressed in a specific scenario forces to do otherwise. Thus, the relative shares between areas for different fruits or vegetables or between chicken and pig production, for example, are assumed to be identical to the reference situation when investigating a shift towards full organic production or reduced concentrate feed use, unless some explicit assumption on those crops or animals is made that requires different assumptions (e.g. when combining reduced concentrate feed use with the goal to use most efficient animals only – thus using pigs rather than chickens for valorizing food waste as feed and in consequence changing the share of pigs to chickens in favor of pigs).

Often, models with an endogenous decision structure are deemed to be more realistic than such exogenously driven models such as SOLm, as the former consistently cover this decision making part of the system based on some underlying societal or psychological theory. However, these models with decision structure also depend on strong assumptions, such as, for economic models, on price elasticities determining how much demand changes with changing prices, or cross-price elasticities determining how much substitution occurs between commodities when their relative prices change. These assumptions are however rather hidden in the model details and not as explicit as the assumptions made exogenously in models such as SOLm. This data (e.g. on elasticities) is by far not always available and often assumptions on the values of these elasticities have to be made on a thin or missing empirical basis. Furthermore, the type of questions addressed with SOLm (primarily large changes, cf. next section 1.4.3) often cannot be addressed well with such models with endogenous decision structure, as the decision structure is calibrated with observed situations and how such decision structure may change with drastic deviations from these calibration points is most often unknown. Economic food system models, for example, are calibrated at the point of general market equilibrium and are most valid in a range around this point, losing validity the further off one may move from this equilibrium in a scenario, as no information is available on how preferences and elasticities may change with such large deviations. Thus, such economic models can well capture effects of an increase of organic production from an observed 15% share to 17% or 20%, or also a decrease to 12%, - but they have not a better empirical basis to answer changes towards 50, 80 or 100% organic production than the models with exogenously given assumptions such as SOLm. Hence, the absence of an endogenous decision structure is not a general disadvantage for the types of questions addressed with these models (see next section 1.4.3).

1.4.3 Which questions can be addressed with SOLm?

SOLm is most adequate to address large changes in the food system and in agriculture with respect to some baseline or reference situation. This is due to its global scope and the default country level resolution of the data used. Further differentiation is always possible, given due data is available, but generally, even with some sub-country regionalization and some differentiation between crop varieties, for example (e.g. winter vs. summer wheat), the results remain on a comparably aggregate and gross level. Thus, given the generally large uncertainties or gross averages and aggregations in the data used, results from SOLm are most helpful when comparing large changes such as a conversion to 30, 50 or 100% organic production rather than smaller changes going from 15 to 17%. Results from such small changes generally will be overshadowed by data uncertainties and error margins. Given all the uncertainties in the data, it is also often more illustrative to assess relative changes between different options than absolute values.

Regarding results, a particular focus of SOLm is to analyse trade-offs and synergies between different indicators for various options or scenarios, or between different strategies or changes with respect to certain indicators, and thus to identify potential options where particular challenges for implementation may arise, or to identify options that are particularly promising. Thus, SOLm results on a large-scale conversion to organic agriculture, for example, illustrate potential trade-offs with sufficient nitrogen supply, and the results on food-competing feed reductions illustrate how drastic the changes in animal source food supply may become and how large changes in animal source food consumption thus may need to be to make such a change in feeding practices and the corresponding implementation of more circular food-systems a widespread option (Schader, Muller et al. 2015, Muller, Schader et al. 2017).

It is important to note that any baseline or reference situation with adequate data availability can be chosen as point of comparison for the options calculated with SOLm. Thus, SOLm results can provide insights on counterfactual situations to the current (or also past) food system, or insights on different options in comparison to, for example, the FAO projections for 2050 from 2012 (Alexandratos and Bruinsma 2012) or in comparison to each of the three updated FAO scenarios for 2050 from 2018 (FAO 2018), or in comparison to the scenarios used in the IPCC report on the 1.5 degree goal (IPCC 2018), etc. If not yet available in SOLm, data for such reference scenarios can be added relatively easily (cf. section 2.4).

Given the absence of any societally or psychologically motivated endogenous decision structure, the results generated by SOLm primarily provide information on the biophysical and agronomic viability of food system options and scenarios with respect to certain biophysical, agronomic or political (or other) restrictions that can be translated in limit or target values for the performance indicators provided by SOLm. The results do not address the socio-economic viability of these options or scenarios in the context of some socioeconomic, psychological or other societal theory. This additional context then provides further restrictions to the viability of options, which are outside the scope of the topics that can be addressed with SOLm. SOLm thus provides a basis for such further assessments and discussions, the main contributions being information on trade-offs and synergies related to different options and indicators and on the options' biophysical and agronomic viability, or their viability regarding some environmental targets. If the latter is not given or very challenging, for example, it can be derived that any further analysis of the respective options may not be needed as these options would anyway be unrealistic due to biophysical or agronomic reasons or due to not reaching certain environmental goals.

SOLm does not work with spatially explicit data and thus cannot address questions that directly rely on processing such data. However, it can clearly make use of information derived from spatially explicit data if the processing of the spatially explicit data is done outside SOLm and the relevant values are then aggregated and fed into SOLm on the level of the geographic regions chosen to run the model (i.e. e.g. on country level). SOLm does neither work on plot or farm level, but as with spatially explicit data, if plot or farm level data is processed outside SOLm and aggregated adequately to be fed into it, SOLm can work with it. See section 2.4 for a description of how to add such and other new data to SOLm.

1.5 Main opportunities and challenges

SOLm uses FAOSTAT (FAO 2019) as its main default data source for cropping areas, animal numbers, yields and production quantities, as well as for trade volumes and domestically available quantities of the commodities and for their utilization shares. IPCC coefficients and data (IPCC 2006) is used as a second main body of data for calculating emissions and impacts. This default data allows for easy linkages of SOLm to a wide range of other analyses using the same data sources, and also allows for simple updates as the data is updated repeatedly and publicly available.

An advantageous flexibility of SOLm is the ease with which new data can be added to refine or overwrite default values (cf. section 2.4). It is easy to read better data for single countries, also with a regional resolution on sub-country level. It is also easy to add new land use or livestock activities (e.g. Miscanthus, which is not covered in FAOSTAT) or to refine existing ones (e.g. separating wheat as from FAOSTAT into summer and winter wheat). Furthermore, data from other studies can be added easily to replicate their results and to use them for consistency checks of SOLm. This applies to the National Greenhouse Gas Inventories submitted by the countries under the UNFCCC (UNFCCC 2019) or to OECD nitrogen and phosphorus balances (OECD 2019), for example. Replicating those is an important consistency check for SOLm baseline calculations (cf. section 2.6).

A disadvantage of the strong reliance on FAOSTAT data as a main default data source is the inconsistency in part of this data. Besides minor misreporting that can be corrected easily when detected (e.g. wrong units by factors of 1000 for certain entries in certain countries, etc.), there is a general problem in a mismatch between animal numbers and the feed quantities reported, where concentrate feed tends to be significantly underreported. This clearly also depends on assumptions on feeding rations, grassland yields and feed supply, animal nutrient requirements, etc., which are described in detail in section 3. With common assumptions on these parameters, these inconsistencies cannot be resolved and are thus related to the original data.

The default procedure to deal with this in SOLm is to explicitly calculate a feed supply/demand ratio in the baseline or reference scenario and to then apply this in all options calculated to assure comparability with the baseline or reference scenario. Thus, if in a certain country FAOSTAT reports only 80% of the feed needed to feed the reported animal numbers, we assume a similar ration in all options/scenarios calculated, i.e. a given quantity of feed can always support 25% more animal feed requirement than its nutrient contents would suggest. This problem arises with the default FAOSTAT data and is also owed to the assumption that data on animal numbers is of better quality than data on feed supply. It is however often resolved when e.g. replacing the default data with specific refined national data. It is also not a problem for all countries. For some countries, this problem is larger, for others, it is nearly absent and feed supply and demand are in balance within a few percent.

An important consequence of this type of inconsistency and its treatment in SOLm is that nutrient flows are not fully closed. A detailed assessment of this is still pending, but somewhere in the system is a gap, e.g. either by reporting exaggerated animal numbers, assuming wrong feeding rations for the animals, reporting too low areas for feed production, or too low feed utilization of the domestically available quantities. Without having tested this further, we hypothesize that data is worst for feeding rations, for feed utilization of domestically available quantities, for the utilization of biomass streams not covered in the data (waste, residues, by-products, etc.) and also for biomass streams from grasslands (where data on areas, yields and production, and also on its nutritional quality as feed are highly uncertain). This mismatch between feed supply and demand is a key issue to be investigated in detail and improved soon, cf. section 2.9.

It is also important to note that for environmental impacts related to feed use in animals, the feed supply/demand ratio is corrected for. Thus, each animal is assigned with its required amount of feed for calculating enteric fermentation emissions, nitrogen excretion or environmental impacts of feed production, irrespective of the aggregate feed supply being less than aggregate feed requirements. This thus again reflects the assumption that animal numbers are largely correct and that the mismatch may rather stem from the feed data.

2. Current Code Structure of SOLmV6

This section describes the structure of SOLmV6 in more detail and explains various aspects related to the use of this model. A detailed description of all data and each code file is then provided in the subsequent section 3.

2.1 Software, platform

SOLmV6 is coded in the Global Algebraic Modelling System GAMS language (as all previous versions of SOLm; we currently use version 24.1 of GAMS) and runs on a separate password-protected server, which is accessible from outside FiBL (vpn- and remote desktop connection, but only for FiBL employees; cf. section 1.2 for information on how to get access to the model). Data is either read from excel-, csv-, gdx-files or directly entered in the GAMS-code. Output is provided in gdx- or excel-files. GAMS can add additional output to existing excel-files without losing the data already stored there. Thus, single model runs can be added to such output files without the need to rerun all previous model runs again. The core code of SOLm does not use features for which GAMS is most adequate, such as optimization routines, but being coded in GAMS, it directly links to such optimization modelling, as e.g. undertaken in other projects where such additional modules are executed on output from SOLm. It is also planned to include the code for optimizations as an additional core part of SOLm.

SOLm is programmed for the Windows Operating System. This can cause problems when trying to run it on MAC OS, for example. Thus, the command “gdxxrw”, for example, does work on Windows only and will produce error messages on the MAC operating systems. Generally, we have no solutions for these problems that arise from differing operating systems, but we mention this here to allow users on other platforms to better identify the causes of certain error messages that may be displayed. For the issue with “gdxxrw”, which is used to read basic data in excel- or csv-format, a workaround is to directly use the corresponding gdx-files only, which are also available on the ftp-server.

2.2 Current structure

Currently, SOLm has a general structure in two main steering files, the first – "`___V6_SteeringFile1_ModelInitialisation.gms`" - governing the filing in and organization of the data for the baseline, as well as of additional data to refine default values, to add new regions or activities, or to add sub-regional or sub-activity level data, as well as to add data for reference scenarios. This file produces several output files, which are then used by the second steering file – "`___V6_SteeringFile2_CoreModelScenariosAndEquations.gms`" – where the scenario definitions and model calculations are done and output files with the results from the model runs are defined and generated.

2.2.1 "___V6_SteeringFile1_ModelInitialisation.gms"

The structure of this file is as follows (the following is the table of contents as used at the beginning of this file):

```
*DETAILED TABLE OF CONTENTS
$ontext;
- 1) General settings
         1.1) Operating systems settings, etc.
         1.2) specify some global variables to chose for global scenario
specifications (yield gap, etc.)
                 1.2.1) specify which yield coefficients to use for organic
                        yields (Badgley, Ponisio, DePonti or Seufert)
                 1.2.2) specify which baseline years to use for the model
                 1.2.3) specify the GWP/GTP to be used
- 2) Specify whether external data needs to be loaded ("YES") (e.g. from
updated data files) or not ("NO")
         2.1) General baseline data
         2.2) Some 2050 scenario data and some 2010 Bioenergy baseline data
         2.3) New regional data
- 3) Load the baseline values for all variables and parameters
         3.1) Define some general sets needed for reading in data
         3.2) Define the general model sets, parameters and variables
         3.3) Read in the data
         3.4) Read refined baseline data
- 4) Initialise the parameters and variables for the model runs
- 5) Add data from FAO and other future scenarios
- 6) Define some output files
$offtext;
```

In more detail:

`- 1) General settings`

This code section defines general operational and key settings to indicate which keys to be used when writing paths to indicate where data is located, etc. (as delimiters in folder structures, for example).

It also allows to choose the values to be used for the organic yield gaps, for the baseline years to be used and for the global warming potential (cf. section 2.3).

`- 2) Specify whether external data needs to be loaded ("YES") (e.g. from updated data files) or not ("NO")`

This part allows to specify whether the data read and reorganized for use in SOLm further down in the code has to be read anew or not. If not read anew, the existing gdx-files of the data are used for reorganizing and making available for use in SOLm. New data has to be read, for example, if FAOSTAT is updated to a new year and this should be used in a new baseline, or if new nutrient contents data is made available, etc. This is then read from excel- or csv-files and stored in gdx-files for further use in the code.

Furthermore, this code part governs the filing in of certain reference scenario data and of other additional data (e.g. for sub-regions) in a similar way as for the baseline data as just described above, i.e. filing it in anew in case the original csv- or excel-files have been changed ("YES"), or using the already available gdx-files if not ("NO").

`- 3) Load the baseline values for all variables and parameters`

This is the largest part in this code-file, governing the filing-in of all baseline values and making them available to SOLm in the SOLm-internal parameter and variable formats. It starts with defining all needed sets (such as sets of activities, regions, inputs, outputs, etc.), parameters (the values for the inputs, etc.) and variables (areas cropped under certain activities, etc.) and then reads the data and translates it to these SOLm parameters and variables.

In this part, code modules for adding new or refined baseline data can be added.

`- 4) Initialise the parameters and variables for the model runs`

This code part already defines and initializes the parameters and variables as used in the model – they are named identical to the respective sets, parameters and variables used for filing in the baseline data, besides a suffix "_MR" that is added to the names (for "model run"). They have an additional last scenario dimension to allow for values to be different for the different scenarios (the baseline values are captured by the scenario "Baseline").

This is needed to read scenario data referring to different future scenarios, such as the reference scenarios from the FAO 2050 projections (FAO 2018), as for these values, we already need a scenario-dimension in the parameters and variables to differentiate them from the baseline.

`- 5) Add data from FAO and other future scenarios`

This part then reads the new scenario data mentioned above; structurally, this works similarly to filing in the baseline data as described above.

`- 6) Define some output files`

This part defines a number of gdx-files containing all the baseline data read, defined and processed in the preceding code. It is organized in a number of specific gdx-files for model sets, parameters and variables.

2.2.2 "___V6_SteeringFile2_ CoreModelScenariosAndEquations.gms"

The structure of this file is as follows (the following is the table of contents as used at the beginning of this file):

```
*DETAILED TABLE OF CONTENTS
$ontext;
- 1) General settings
        1.1) Operating systems settings, etc.
        1.2) specify some global variables to choose for global scenario
```

```
                 assumptions (how to allocate mineral fertilizer, etc.)
- 2) Define sets, parameters and variables and load gdx files from the
            baseline assignment
- 3) Run core model equations
- 4) Choice of scenarios
- 5) Further calculations after finishing the scenario runs
- 6) Define some output files
- 7) Do some further specific calculations needed for certain aspects
$offtext;
```

In more detail:

- 1) General settings

This code section defines general operational and key settings to indicate which keys to be used when writing paths to indicate where data is located, etc. (as delimiters in folder structures, for example).

It also allows to choose the way how to allocate mineral fertilizers (cf. section 2.3).

Finally, it allows to choose whether only one scenario is run or whether the code is executed as a loop over a number of scenarios, as defined towards the end of the steering file 2 (governed by `$setglobal RunAllChosenScenarios "NO"` or `$setglobal RunAllChosenScenarios "YES"`)

- 2) Define sets, parameters and variables and load gdx files from the baseline assignment

This part defines all the sets, parameters and variables needed in the model in general and for filing in the data compiled in "`___V6_SteeringFile1_ModelInitialisation.gms`" in particular and loads all the data compiled there. It also initializes the model run parameters and variables with the baseline values.

- 3) Run core model equations

This is the largest part in this code-file, governing the core model calculations.

It starts with a module initializing the scenario data with baseline or reference scenario values, and then setting the scenario specifications where specific scenario-related values are available. These assumptions in particular provide data on the areas for each crop and grassland type etc. in the scenario.

It then derives the total crop production and domestically available quantities (DAQ) for crops and grass. For the DAQ values, default import/export relations as in the baseline or reference scenario are assumed and the trade flows are scaled proportionally to changes in production (i.e. exports scale with domestic production, imports scale with production in the country of origin). Thereby, trade flows are traced back to the original country of production (cf. section 3.12). This allows to deal e.g. with the case that Germany imports large amounts of Soybeans from The Netherlands – which are themselves imported from Brazil – thus, this data-reorganization allows to show that Germany in fact imports from Brazil.

Then some data on crop residue management are calculated, as well as on crop and grassland nutrient requirements, animal nutrient requirements and feed supply.

Based on this, animal numbers and animal production, DAQ for animal source commodities, trade in animal commodities, manure excretion and enteric fermentation are derived and aspects related to manure management are modelled.

Finally, fertilizer application (mineral fertilizers, crop residues and manure) and related emissions are calculated.

- 4) Choice of scenarios

Here, it is defined which scenarios shall be run in which order. The first run is for the "BaselineDerived", set at the beginning of the model code section "`3) Run core model equations`", by means of the code line "`$setglobal Scenario "BaselineDerived`". This replicates the baseline scenario and produces a number of additional output. This is also a test for the code, as it should be identical to the scenario "Baseline" on all indicators, where values for this and for "BaselineDerived" are available. After running all code files in section "`3) Run core model equations`", the scenario name is set to the next scenario of interest and the code jumps at the beginning of "`3) Run core model equations`" and executes this again. This loop is repeated until the last scenario name triggers "`EndOfScenarioRuns`", with which the execution continues in section "`5) Further calculations after finishing the scenario runs`".

An example of such a scenario choice set is displayed here:

```
$setglobal Scenario "BaselineDerived"
$label Restart
```

Then comes all code in section "`3) Run core model equations`"

Then comes the choice in section "`4) Choice of scenarios`", for example the following, which would first calculate a scenario, where the baseline would be converted to 100% organic production and then a scenario, where no food-competing feed is used:

```
$if %Scenario% == "Baseline_NoFCF" $goto EndOfScenarioRuns
$if %Scenario% == "Baseline_100Organic" $setglobal Scenario "Baseline_NoFCF"
$if %Scenario% == "BaselineDerived" $setglobal Scenario "Baseline_100Organic"

$if %RunAllChosenScenarios% == YES $goto Restart

$label EndOfScenarioRuns
```

- 5) Further calculations after finishing the scenario runs

After the scenario runs, a number of impact indicators are derived, on a per unit basis (per hectare, per unit commodity (mass, protein, calories, economic value), per head, per animal production unit) and on aggregate. This includes also an allocation between co-products, based on mass, on protein or calorie contents, or on economic value. Aggregations of various variables over regions and activities are calculated.

- 6) Define some output files

This part defines a number of gdx-files containing all the results produced in the preceding code. It is organized in a number of gdx-files for model sets, parameters and variables (generated in "`_V6_OutputFiles_SteeringFile2.gms`"), as well as some specific result file providing data as a basis for further processing, depending on which analysis the model runs aim at. This processing thus can include graphics, tables, deriving further aggregate values, etc. and is done outside SOLm. This compilation of specific results is produced in the code file "`_V6_ResultsFiles.gms`".

`- 7) Do some further specific calculations needed for certain aspects`

Currently, no code is included here, but this would be the location to easily add some code for some further calculations, such as e.g. to run certain comparisons, test certain values, compile special output formats, etc.

2.3 Possible parameter choices of general model validity

This section shortly describes the parameter choices that are done with general model validity and normally do not change for single scenarios, albeit it is possible to also overwrite these choices in the scenario definitions, if needed.

For "`___V6_SteeringFile1_ModelInitialisation.gms`", in section "`- 1) General Settings`" the following choices can be made: baseline years; organic yield gaps; global warming/temperature potential; some assumptions for bioenergy: sections 2.3.1 to 2.3.4.

For "`___V6_SteeringFile2_ CoreModelScenariosAndEquations.gms`" the following choice can be made: allocation of mineral fertilizers section 2.3.5.

2.3.1 Baseline years

The baseline is usually an average over a number of years and refers to these average values taken from FAOSTAT. Here in the code, it is indicated over which years this average shall be taken. Flexibility in this is large, any time-period previous or including the last year where data is available is possible. The code for calculating the baseline values is – where applicable – executed each time right after filing in new data in the respective code modules. It thereby accounts for missing values when deriving the averages by correspondingly correcting the numbers of years to be divided by.

2.3.2 Organic yield gaps

Organic yield gaps can be chosen from four sources, namely (Badgley, Moghtader et al. 2007, de Ponti, Rijk et al. 2012, Seufert, Ramankutty et al. 2012) and (Ponisio, Gonigle et al. 2014). The default is (Seufert, Ramankutty et al. 2012) or (Ponisio, Gonigle et al. 2014), from Summer 2020 onwards, usually the latter is used (Ponisio). Furthermore, for the Seufert2012- and Ponisio2014-data it is possible to indicate, whether organic yield gaps in developing countries are identical to those in developed countries or whether yield gaps in developing countries should be zero. This choice is available due to lack of information on yield gaps in developing countries and the hypothesis supported by some that yield gaps in developing countries are lower than in developed countries. Generally, we tend to conservatively assume that yield gaps are similar and set this choice of similar yield gaps as the default. For Badgley2007-data, it is possible to choose organic yields in developing countries to be higher than conventional yields, as suggested by their data. This is however highly contested and the default in SOLm is to not do so. In any case, (Badgley, Moghtader et al. 2007) is a highly contested publication and we suggest to not use their yield gap data.

Currently, an update to the most recent most encompassing review on organic yield gaps provided in (Seufert 2018) has to be done. Furthermore, one could attempt to derive yield gaps from the area and production data as reported in the World of Organics (REFERENCE).

2.3.3 GWP and GTP

It is also possible to choose a number of values for the global warming potential (GWP) of different greenhouse gases, or to choose to use global temperature potential (GTP) rather than GWP. Currently, only data for different GWP-values are available in the code, but other values and GTP values can be added easily. The current choices differ by source ((Ramaswamy, Boucher et al. 2001) as used in the IPCC 2006 Guidelines on GHG Inventories or from IPCC AR5 (Myhre, Shindell et al. 2013)) and, for methane, by accounting or not for the fact that most methane in agriculture is biogenic, which results in slightly lower values, as the final decay product of methane – CO_2 – is renewable in this case.

The update to IPCC (2019 – check: new GWP data?) or including GWP* to better capture the effects of the relatively short lifetime of methane (REFERENCE) have still to be implemented.

2.3.4 Some assumptions for bioenergy

Some assumptions on bioenergy are currently also governed in this part of the code. They refer to assumptions on the nitrogen fertilization level of Miscanthus, i.e. on whether this is low, medium or high, and on the nitrogen contents of the bioenergy residues after bioenergy production, which are assumed to be low or high. These are very specific assumptions and may rather be moved to scenario specifications in a further revision of the model code. TO BE DONE!

2.3.5 Allocation of mineral fertilizers

This assumption governs globally, whether mineral fertilizers are applied to crops on croplands only or to temporary grasslands as well.

2.4 Adding new data

In SOLmV6, the baseline data is read in "___V6_SteeringFile1_ModelInitialisation.gms". Towards the end of this file, also some reference scenario data for the most recent FAO projections towards 2050 (FAO 2018) and for some scenarios from the IPCC 1.5-degree goal report (IPCC 2018) are read.

In this version, new data for both baseline values or also new reference scenarios is then added via additional code files to be run from "___V6_SteeringFile1_ModelInitialisation.gms", just after having read the original data. Thus, for baseline data it is currently to be added and executed after the module "_V6_ReadData_VariousSources_CED" and before "_V6_DataDerivedBaseline_SomeHerdStructureParameters"; for the reference scenario data it would be again in "___V6_SteeringFile1_ModelInitialisation.gms" but after "_V6_ReadData_VariousSources_BioenergySR15.gms" and before "_V6_StreamlineInitialData.gms".

Currently, modules for reading better animal number data for the EU, detailed data on GHG inventories (currently from Austria and Switzerland), better price data for Switzerland and better feed data for Switzerland are available.

It is planned to reorganize this slightly and to collect the adding of new data into an additional separate steering file to be executed with the outputs from "___V6_SteeringFile1_ModelInitialisation.gms" and delivering as outputs all inputs to "___V6_SteeringFile2_ CoreModelScenariosAndEquations.gms" (cf. section 2.7).

New data can also be added in the scenario definitions. This is the approach followed for data which need not enter the general model data base of SOLm but rather be used in specific scenario runs only, as it can be done then without running "___V6_SteeringFile1_ModelInitialisation.gms". However, in principle, all new data, be it also very specific and restricted in application, should at some point of time be added to the general model data base for later availability.

Basically, for the baseline, new data just overwrites the existing data. If data on sub-regions or sub-activities is added, these sub-regions and sub-activities are first initialized with the data from the corresponding regions and activities as far as relative values (per hectare, per animal head, per ton output, etc.) are concerned. If better data on these values is available, this overwrites this default assignment. Absolute values (e.g. areas per crop, etc.) are assigned with new better data, if available, or with data derived from the corresponding regional or activity-specific total values, by means of some proportionality rules. There is, for example, rarely data on import and export values to sub-regions available. Thus, regional trade data is per default assigned to sub-regions proportional to production (exports) or population (imports), for example.

New data can also be added in closer relation to some complementary modelling, such as grid-based spatially explicit assessments or plot- or farm-models, in detail tracing nutrient- and mass-flows on smaller scales than captured by the spatial or activity-related resolution of SOLm. Linkage to SOLm is then established by taking averages and by aggregating the results from these finder models to a level that can be used as input to SOLm (e.g. country or activity level).

Further details on adding new data are given in section 3.5.

2.5 Model output

Output from the baseline data processing is produced at the end of file "_V6_OutputFiles_SteeringFile1.gms". This output is used as an input to "___V6_SteeringFile2_ CoreModelScenariosAndEquations.gms". This latter file then generates the general model output. It has two parts. First, it is in the form of several files containing all sets, parameter and variables that are used in the model. These result files are generated in "_V6_OutputFiles_SteeringFile2.gms". Second, there is a compilation of results for further processing, depending on which analysis the model runs aim at, including graphics, tables, aggregate values, etc. This compilation of specific results is produced in the code file "_V6_ResultsFiles.gms" and can be adapted to the specific needs one may have for a specific project.

2.6 Consistency checks

Consistency checks in SOLmV6 are mainly done by comparing results to results on the same variables or parameters from other sources. Basically, such comparisons can be done on a relative basis (e.g. comparing emission factors per kg commodity) or on an aggregate basis (e.g. comparing total GHG emissions from the livestock sector in a country). There is a wealth of data available that should in principle be replicable by SOLm – or where it should be possible to understand any deviations very well – and that can thus serve as a basis for such checks. Examples are LCA databases for footprints of various commodities regarding a large number of indicators, or the national UNFCCC GHG inventories or the national OECD nutrient balances for aggregate values.

A second type of consistency check relate to internal consistency of different nutrient flows, e.g. checking whether the total nitrogen applied per hectare corresponds to the total extraction from it (inclusive losses), duly accounting for the possibility to add or extract nitrogen to or from the soil pool. This latter is currently not covered in the model, but doing this balancing of nutrient flows from inputs and outputs and losses allows to derive an approximate size of such soil-pool related flows and the consistency check then consists in assessing whether they have a reasonable size or not, for example. Similarly, such balancing can be done for animals with their feed uptake and manure extraction, etc. or for manure management, with incoming nutrient quantities from manure, outgoing nutrient quantities after management for application on fields, and management losses. TO BE DONE

Currently, consistency checks are done by manually comparing some key indicators for a number of regions and activities or aggregates thereof to established literature values (e.g. the GHG inventories and OECD-N-balances have been checked in detail for Austria and Switzerland – and they are replicated very well; see specific excel-file SOLm_ConsistencyChecks.xlsx in the subfolder DataForConsistencyChecks on the ftp-Server where the data for SOLm is stored), and by doing some specific balances and comparing with expected literature values (e.g. for soil-pool storage of N). It is planned to make this process more encompassing and automatized. For further details see section 3.14.

2.7 Known code issues to be improved next

In the following, for completeness, some next steps for improvement of the code are listed.

First, it is planned to reorganize slightly how new data is read and to collect the corresponding code into an additional separate steering file to be executed with the outputs from the current code file "___V6_SteeringFile1_ModelInitialisation.gms" and delivering as outputs all inputs needed to execute the current code file "___V6_SteeringFile2_ CoreModelScenariosAndEquations.gms".

Thus, "___V6_SteeringFile1_ModelInitialisation.gms" would be shortened to execute all code modules as now besides the current four files with new data:

"_V6_ReadAdditionalData_AnimalNumbers.gms"

"_V6_ReadAdditionalData_GHGInventories.gms"

"_V6_ReadAdditionalData_Prices.gms"

"_V6_ReadAdditionalData_FeedData.gms"

(see section 3 for details).

A new second steering file would then be coded exclusively for adding new data, e.g. "___V6_SteeringFile2_AddingNewData.gms". This would start with the code needed to read all output from "___V6_SteeringFile1_ModelInitialisation.gms", as coded now at the beginning of "___V6_SteeringFile2_CoreModelScenariosAndEquations.gms". Then new baseline or scenario data would be added (i.e. e.g. the file named above). Then the modules "_V6_DataDerivedBaseline_SomeHerdStructureParameters.gms" (doing some herd-structure parameters that can only be done after having read new data) and "_V6_StreamlineInitialData.gms" (doing some streamlining of the new data, if needed) – for details on these code files, see section 3 – and the code producing the output files (analogue to the current file "_V6_OutputFiles_SteeringFile1.gms") to be fed into the now third steering file "___V6_SteeringFile3_ CoreModelScenariosAndEquations.gms" would be executed.

In principle, the reference scenario data currently read in the first steering file in the code files "_V6_ReadData_FAOSTAT_FOFA2050.gms" and "_V6_ReadData_VariousSources_BioenergySR15.gms" could also be moved to the second steering file on filing in new data.

Then, the new third steering file ("___V6_SteeringFile3_ CoreModelScenariosAndEquations.gms") would be run as is currently the second one.

Second, the general parameter choices on "organic yield gaps"; "global warming/temperature potential"; "some assumptions for bioenergy"; "allocation of mineral fertilizers"; currently executed at the beginning of the two steering files could be decoupled from those and rather added to the scenario definitions than to the basic data. Thus, no changes in the basic data would be needed in case one would like to change these parameter choices. To do this, some reorganization of the code would however be necessary, e.g. to keep the yield gaps from the different sources separate in the basic data in SOLm parameters – currently, there is only one parameter for yield gap which is assigned the respective value conditional to this general parameter choice at the beginning of the steering file. Similar adjustments would be needed for "global warming/temperature potential"; "some assumptions for bioenergy". For "allocation of mineral fertilizers" from the second steering file this is different and it could be moved to the scenario definitions without further adjustments needed, as this file governs the choice of one or another part in the calculations, without assignments of data. As the code is structured now, the settings on the baseline choice could not be moved to the scenarios without fundamental recoding – but this is neither a big problem, as all scenario runs usually refer to the same baseline. And in the context of one project, the baseline usually does not change.

Third, the herd structure model should be updated, expanded to chickens and re-run and linked to SOLmV6, as well as made consistent with more recent developments at FAOSTAT, such as the GLEAM model. Currently, the values from the calculations done for (Schader, Muller et al. 2015) are used (cf. section 3.1.43), but for the EU, better data from Eurostat is used (cf. section XXXXX – read additional data animal numbers).

Fourth, the animal production unit (APU) – view on the animal production (cf. the description towards the end of section 1.3) should be coded systematically (i.e. by providing the needed sets, parameters and variables to implement this approach consistently). If this makes sense in the baseline, it would mean to do the hitherto empty code module "_V6_ReadData_VariousSources_AnimalProductionUnits.gms", see section 3.1.44. But it could also be added on the level of Steering File 2. Currently, APUs play a role only towards the end of Steering File 2 for the impact calculations for animal source commodities, as their impacts are based on the impacts of the whole herd and not of the producing animals only (cf. section 3.2.23).

Fifth, the consistency checks should be made automatized, such as to have a default set of values to run comparisons for, e.g. after introducing new and refined data. Furthermore, consistency checks for nutrient flows on a per area and per animal head basis should be coded.

Sixth, the currently independent optimization code used in some projects and building on output from SOLm should be added as an integral part to SOLm.

2.8 Next things to be added

First, there is a number of smaller things to be added in the baseline data in "___V6_SteeringFile1_ModelInitialisation.gms". They are directly highlighted in the code. They are planned to be addressed in the coming 3-4 months. In particular, the most recent data from (Seufert 2018) on yield gaps should be added and made the default to be used in SOLm. The potential to use the data from World of Organics (REFERENCE) to derive yield gaps should also be explored.

Furthermore, GWP*-data should be added to be able to differently account for methane. And the reference scenario data from FAO (2018) and the SR1.5 IPCC (2018) should be added.

Bigger topics that are partly currently addressed by master theses are

- bioenergy
- fish, seafood and aquaculture
- alternative protein sources
- vegan food systems

For each of those, encompassing databases should be compiled, allowing for scenarios to be run with a focus on various aspects of these respective topics.

Finally, a thorough check of the new IPCC guidelines IPCC (2019) should be done and the model code adapted to it where necessary.

2.9 Open questions and inconsistencies to be addressed next

The most pressing open question currently in SOLm is the mismatch between feed supply/demand mentioned above. It is necessary to dig into this deeper to better understand which data in FAOSTAT (e.g. from the feed utilization shares, domestically available quantities for feed, feed streams not covered such as residues, by-products or waste for feed, or other data) or elsewhere (e.g. from the feeding rations data) causes these inconsistencies.

Furthermore, the grassland data has to be cross-checked again, besides areas and yields, which has been updated recently, in particular the feed quality of grasslands, legume shares, protein and nitrogen contents, etc.

3. SOLmV6: Code and Data in Detail

This section presents the code in detail, going through all code modules that are executed in the course of running the two steering files as described in section 2.2 on the general model structure (sections 3.1.1 to 3.2.28). After this some specific aspects of the model code are addressed in further detail, such as how to specify scenarios, how to add new data, how to produce results files, how to treat crop rotations and herd structures, how to organize feed supply, how to capture bioenergy or fish and seafood, etc. (sections 3.3 to 3.14). Finally, some details on some of the data used are presented (sections 3.15 to **Fehler! Verweisquelle konnte nicht gefunden werden.**).

3.1 SteeringFile 1

The following describes the content of the code files executed in "___V6_SteeringFile1_ModelInitialisation.gms" in some detail. The general structure of this file is described in section 2.2.1. In the following, we shortly list all code modules that are executed and subsequently describe those in detail (the headings displayed are the same as in the structure described in section 2.2.1):

Code modules executed in Steering File 1:

```
3) Load the baseline values for all variables and parameters
        3.1) Define some general sets needed for reading in data
```

_V6_Sets_FAOSTAT_Regions
_V6_Sets_FAOSTAT_Items
_V6_Sets_FAOSTAT_ItemGroups
_V6_Sets_NonFAOSTAT_Items
_V6_Sets_FAOSTAT_Elements
_V6_Sets_FAOSTAT_Units
_V6_Sets_FAOSTAT_LandUse
_V6_Sets_FAOSTAT_Deforestation
_V6_Sets_FAOSTAT_OrganicSoils
_V6_Sets_ErbEtAl_Grasslands
_V6_Sets_FAOSTAT_Fertilizers
_V6_Sets_FAOSTAT_Population_HumanNutrReq
_V6_Sets_VariousSources_HerdStructures
_V6_Sets_GeneralModelSets_ForReadingData

```
        3.2) Define the general model sets, parameters and variables
```

_V6_Sets_GeneralModelSets
_V6_VariablesAndParameters

```
        3.3) Read in the data
```

_V6_ReadData_FAOSTAT_CropProduction
_V6_ReadData_FAOSTAT_ForageCropProduction
_V6_ReadData_FAOSTAT_LivestockProduction
_V6_ReadData_FAOSTAT_Trade
_V6_ReadData_FAOSTAT_CommodityBalances
_V6_ReadData_FAOSTAT_LandUse
_V6_ReadData_FAOSTAT_Deforestation
_V6_ReadData_FAOSTAT_OrganicSoils
_V6_ReadData_ErbEtAl_Grasslands
_V6_ReadData_FAOSTAT_Fertilizers
*_V6_ReadData_FAOSTAT_WOSY_DetailedFBS
This file is not used anymore
_V6_ReadData_FAOSTAT_Population
_V6_ReadData_VariousSources_HumanNutrientRequirements
_V6_ReadData_VariousSources_CropGrassNutrientRequirementsData
_V6_ReadData_VariousSources_MainOutputNutrientContentsData
_V6_ReadData_VariousSources_ResidueSharesAndNutrientContentsData
_V6_ReadData_VariousSources_SeedCharacteristicsData
_V6_ReadData_FAOSTAT_ProducerPrices
_V6_ReadData_IPCC_GWP_GTPData
_V6_ReadData_LuEtAl_NDepositionData
_V6_ReadData_VariousSources_NFixationData
_V6_ReadData_VariousSources_SoilErosionData
_V6_ReadData_VariousSources_IrrigationWaterData

_V6_ReadData_VariousSources_AnimalWelfareData
_V6_ReadData_VariousSources_PesticidesData
_V6_ReadData_IPCC2006_RiceCroppingEmissionsData
_V6_ReadData_VariousSources_HerdStructures
_V6_ReadData_VariousSources_AnimalProductionSystems
_V6_ReadData_VariousSources_AnimalLiveweightData
_V6_ReadData_VariousSources_AnimalDrinkingWaterRequirementData
_V6_ReadData_VariousSources_FeedingRationsData
_V6_ReadData_VariousSources_AnimalNutrientRequirementsData
_V6_ReadData_VariousSources_EntericFermentationEmissionsData
_V6_ReadData_VariousSources_CropResidueManagementData
_V6_ReadData_VariousSources_ManureExcretionData
_V6_ReadData_VariousSources_ManureManagementData
_V6_ReadData_VariousSources_MineralFertilizerProductionEmissionsData
_V6_ReadData_VariousSources_FertilizerApplicationData
_V6_ReadData_VariousSources_NH3Emissions
_V6_ReadData_VariousSources_OrganicYieldGapsData
*__SOLmV5_DataDerivedBaseline_DetailedFeedingRations

This file is not used anymore

_V6_ReadData_VariousSources_CED

3.4) Read refined baseline data

_V6_ReadAdditionalData_SwitzerlandAustria

This file is not used anymore

_V6_ReadAdditionalData_SwitzerlandAustriaFeed

This file is not used anymore

_V6_ReadAdditionalData_AnimalNumbers
_V6_ReadAdditionalData_GHGInventories
_V6_ReadAdditionalData_Prices
_V6_ReadAdditionalData_FeedData
_V6_ReadAdditionalData_NUTS2_EU

Not yet coded

_V6_DataDerivedBaseline_SomeHerdStructureParameters
_V6_ReadData_CommodityTrees_LinkActivitiesAndCommodities

4) Initialise the parameters and variables for the model runs

_V6_VariablesAndParameters_ModelRun

5) Add data from FAO and other future scenarios

_V6_ReadData_FAOSTAT_FOFA2050
_V6_ReadData_VariousSources_BioenergySR15
_V6_StreamlineInitialData

6) Define some output files

_V6_OutputFiles_SteeringFile1

The following subsections provide detailed descriptions of these code modules.

3.1.1 _V6_Sets_FAOSTAT_Regions

This file declares the sets where all countries, geographic regions and aggregates thereof as used in FAOSTAT are collected and it defines the hierarchical structure of those. The sources where these sets come from and partly some further information are given at the beginning of each set definition in the code.

In this code file, the following sets are defined:

FAOSTAT_GeographicRegionsCode

    Country and region codes that may be used in FAOSTAT

FAOSTAT_CountriesAndRegions

    Country and region names that may be used in FAOSTAT

FAOSTAT_CountriesAndRegionsWithCode(FAOSTAT_GeographicRegionsCode,FAOSTAT_CountriesAndRegions)

    Two dimensional set containing all country and region codes and names matched that may be used in FAOSTAT

FAOSTAT_Countries(FAOSTAT_CountriesAndRegions)

    Sub-set containing all country names that may be used in FAOSTAT

FAOSTAT_Regions(FAOSTAT_CountriesAndRegions)

    Sub-set containing all region names that may be used in FAOSTAT

FAOSTAT_Continents(FAOSTAT_Regions)

    Sub-set containing all continent names that may be used in FAOSTAT

FAOSTAT_Subcontinents(FAOSTAT_Regions)

    Sub-set containing all sub-continent names that may be used in FAOSTAT

FAOSTAT_DevelopedDevelopingGroups(FAOSTAT_Regions)

    Sub-set containing developed-developing country groups that may be used in FAOSTAT

FAOSTAT_ContinentsSubcontinents(FAOSTAT_Regions,FAOSTAT_CountriesAndRegions)

    Two dimensional set containing all sub-continents matched to continents

FAOSTAT_CountriesInRegions(FAOSTAT_Regions,FAOSTAT_CountriesAndRegions)

    Two dimensional set containing all regions matched to countries

3.1.2 _V6_Sets_FAOSTAT_Items

This file declares the sets in which ALL activity and commodity items as used for reading in the original data from FAOSTAT are contained. The sources where these sets come from and partly some further information are given at the beginning of each set definition in the code.

In this code file, the following sets are defined:

FAOSTAT_ItemCode

    All crop item codes that may be used in FAOSTAT

FAOSTAT_CropProductionItems

    All crop item names that may be used in FAOSTAT

FAOSTAT_CropProductionItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_CropProductionItems)

    Two dimensional set containing all crop item codes and names matched that may be used in FAOSTAT

FAOSTAT_CropsProcessedItems

    All processed crop item names that may be used in FAOSTAT

FAOSTAT_CropsProcessedItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_CropsProcessedItems)

    Two dimensional set containing all processed crop item codes and names matched that may be used in FAOSTAT

FAOSTAT_LivestockPrimaryItems

    All livestock primary item names that may be used in FAOSTAT

FAOSTAT_LivestockPrimaryItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_LivestockPrimaryItems)

    Two dimensional set containing all livestock primary item codes and names matched that may be used in FAOSTAT

FAOSTAT_LiveAnimalsItems

    All live animal item names that may be used in FAOSTAT

FAOSTAT_LiveAnimalsItems_Poultry(FAOSTAT_LiveAnimalsItems)

    Sub-set containing all live poultry animal item names that may be used in FAOSTAT

FAOSTAT_LiveAnimalsItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_LiveAnimalsItems)

    Two dimensional set containing all live animal item codes and names matched that may be used in FAOSTAT

FAOSTAT_LivestockProcessedItems

All processed livestock item names that may be used in FAOSTAT

FAOSTAT_LivestockProcessedItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_LivestockProcessedItems)

Two dimensional set containing all processed livestock item codes and names matched that may be used in FAOSTAT

FAOSTAT_DetailedTradeMatrixItems

Commodity item names as used in the detailed trade matrix of FAOSTAT

FAOSTAT_DetailedTradeMatrixItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_DetailedTradeMatrixItems)

Two dimensional set containing all detailed trade commodity item codes and names matched that may be used in FAOSTAT

FAOSTAT_TradeLiveAnimalsItems

Live animal item names as used in the live animal trade matrix of FAOSTAT

FAOSTAT_TradeLiveAnimalsItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_TradeLiveAnimalsItems)

Two dimensional set containing all live animal trade item codes and names matched that may be used in FAOSTAT

FAOSTAT_TradeCropsAndLivestockProductsItems

Crop and livestock product item names as used in the general trade matrix of FAOSTAT

EighteenItemsInDetTradeItems_NotInTradeProdItems

18 items (mainly live animals) covered in the set FAOSTAT_DetailedTradeMatrixItems but not in FAOSTAT_TradeCropsAndLivestockProductsItems

FAOSTAT_TradeCropsAndLivestProdItems_NODoubles

Set with items in FAOSTAT_TradeCropsAndLivestockProductsItems that are NOT included in FAOSTAT_LivestockFishCommodityBalancesItems and FAOSTAT_CropCommodityBalancesItems and NON_FAOSTAT_CropProductionItems

FAOSTAT_TradeCropsAndLivestockProductsItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_TradeCropsAndLivestockProductsItems)

Two dimensional set containing all trade crop and livestock item codes and names matched that may be used in FAOSTAT

FAOSTAT_CropCommodityBalancesItems

Set containing the commodity balances crop items

FAOSTAT_CropCommodityBalancesItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_CropCommodityBalancesItems)

Two dimensional set containing all commodity balances crop item codes and names matched that may be used in FAOSTAT

FAOSTAT_LivestockFishCommodityBalancesItems

Commodity balances livestock and fish items

FAOSTAT_LivestockFishCommodityBalancesItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_LivestockFishCommodityBalancesItems)

Two dimensional set containing all commodity balances livestock and fish item codes and names matched that may be used in FAOSTAT

FAOSTAT_ProducerPriceItems

Crop and livestock items for producer price data

FAOSTAT_ProducerPriceItemCodeAndItems(FAOSTAT_ItemCode,FAOSTAT_ProducerPriceItems)

Two dimensional set containing all producer price data crop and livestock item codes and names matched that may be used in FAOSTAT

3.1.3 _V6_Sets_FAOSTAT_ItemGroups

This file declares the sets for groups of activity and commodity items as defined in FAOSTAT.

In this code file, the following sets are defined:

FAOSTAT_CropProductionItemGroups

Crop item groups as included in FAOSTAT

FAOSTAT_CropProductionItemGroups_CompleteExclusive(FAOSTAT_CropProductionItemGroups)

Sub-set containing mutually exclusive crop item groups as included in FAOSTAT

FAOSTAT_CropProductionItemsInItemGroups(FAOSTAT_CropProductionItemGroups,FAOSTAT_CropProductionItems)

Two dimensional set containing all crop groups matched to single crop items

FAOSTAT_LivestockPrimaryItems_Meat(FAOSTAT_LivestockPrimaryItems)

Sub-set containing MEAT items only in livestock primary items

FAOSTAT_LivestockPrimaryItems_PoultryMeat(FAOSTAT_LivestockPrimaryItems)

Sub-set containing POULTRY MEAT items only in livestock primary items

FAOSTAT_LivestockPrimaryItems_Milk(FAOSTAT_LivestockPrimaryItems)

Sub-set containing MILK items only in livestock primary items

FAOSTAT_LivestockPrimaryItems_Eggs(FAOSTAT_LivestockPrimaryItems)

Sub-set containing EGG items only in livestock primary items

FAOSTAT_LivestockPrimaryItems_HidesSkinsHair(FAOSTAT_LivestockPrimaryItems)

Sub-set containing Hides-Skins-Hair items only in livestock primary items

The following files refer to item groups used in earlier versions of SOLm (up to V4 and for reading in some data in V6):

SOLmOld_CropProductionItemGroups

OLD crop item groups as used in earlier versions of SOLm (up to V4 and for reading in some data in V5)

SOLmOld_Fruits(FAOSTAT_CropProductionItems)

FRUIT crop items building the OLD SOLm fruit group

SOLmOld_Treenuts(FAOSTAT_CropProductionItems)

TREENUT crop items building the OLD SOLm treenut group

SOLmOld_Grains(FAOSTAT_CropProductionItems)

GRAINS crop items building the OLD SOLm grains group

SOLmOld_Pulses(FAOSTAT_CropProductionItems)

PULSES crop items building the OLD SOLm pulses group

SOLmOld_Oil_Crops(FAOSTAT_CropProductionItems)

OIL CROP crop items building the OLD SOLm oil crop group

SOLmOld_Starchy_Roots(FAOSTAT_CropProductionItems)

STARCHY ROOTS crop items building the OLD SOLm starchy roots group

SOLmOld_Sugars_And_Sweeteners(FAOSTAT_CropProductionItems)

SUGARS AND SWEETENERS crop items building the OLD SOLm sugars and sweeteners group

SOLmOld_Vegetables(FAOSTAT_CropProductionItems)

VEGETABLES crop items building the OLD SOLm vegetables group

SOLmOld_Stimulants_Spices(FAOSTAT_CropProductionItems)

STIMULANTS AND SPICES crop items building the OLD SOLm stimulants and spices group

SOLmOld_Fibres_Rubber(FAOSTAT_CropProductionItems)

FIBRES AND RUBBER crop items building the OLD SOLm fibres and rubber group

SOLmOld_ForReadingDataBerries(FAOSTAT_CropProductionItems)

BERRIES crop items building the OLD SOLm berries group

SOLmOld_ForReadingDataNonLeguminousVegetables(FAOSTAT_CropProductionItems)

NON-LEGUMINOUS VEGETABLES crop items building the OLD SOLm non-leguminous vegetables group

SOLmOld_ForReadingDataLeguminousVegetables(FAOSTAT_CropProductionItems)

LEGUMINOUS VEGETABLES crop items building the OLD SOLm leguminous vegetables group

3.1.4 _V6_Sets_NonFAOSTAT_Items

This file declares additional sets for activity and commodity items that are currently not covered under FAOSTAT, but have once been (fodder crops) or are other most basic amendments to FAOSTAT items (algae, etc).

In this code file, the following NON-FAOSTAT sets are defined:

NON_FAOSTAT_CropProductionItems
: Additional crop item names that may be used and are NOT part of FAOSTAT

NON_FAOSTAT_GrassActivities(NON_FAOSTAT_CropProductionItems)
: Sub-set containing all grass item names that may be used and are NOT part of FAOSTAT

NON_FAOSTAT_LivestockPrimaryItems
: Additional livestock primary item names that may be used and are NOT part of FAOSTAT

NON_FAOSTAT_LiveAnimalsItems
: Additional live animal item names that may be used and are NOT part of FAOSTAT

NON_FAOSTAT_TradeLiveAnimalsItems
: Additional live animal item names that may be used in the live animal trade matrix but are NOT part of FAOSTAT

NON_FAOSTAT_LivestockFishCommodityBalancesItems
: Additional commodity balances livestock and fish items that are NOT part of FAOSTAT

3.1.5 _V6_Sets_FAOSTAT_Elements

This file declares all the sets with elements needed in the model for reading FAOSTAT production, trade food balance and land use data. “Elements” are, grossly speaking” parameter and variables of interest, such as “yield” or “import quantity” (for further examples see below).

In this code file, the following sets are defined:

FAOSTAT_ElementCode
: All element codes that may be used in FAOSTAT

FAOSTAT_CropProductionElements
: All crop production elements (such as "area harvested" or "yield") that are used in FAOSTAT

FAOSTAT_LivestockProductionElements
: All livestock production elements (such as "Stocks" or "yield") that are used in FAOSTAT

FAOSTAT_TradeElements All trade elements (such as "Import Quantity") that are used in FAOSTAT

FAOSTAT_CommodityBalancesElements
: All commodity balances elements (such as "Domestic supply quantity" or "Waste") that are used in FAOSTAT

FAOSTAT_ProducerPriceElements
: All producer price elements (such as "Producer Price") that are used in FAOSTAT

3.1.6 _V6_Sets_FAOSTAT_Units

This file declares all the sets with units needed in the model for reading FAOSTAT production, trade, food balance and land use data.

In this code file, the following sets are defined:

FAOSTAT_CropProductionUnits
: All units (such as "ha" or "tonnes") that may be used in crop production in FAOSTAT

FAOSTAT_LivestockProductionUnits
: All units (such as "Head" or "hg") that may be used in livestock production in FAOSTAT

FAOSTAT_TradeUnits
: All units (such as "Head" or "tonnes") that may be used in trade data in FAOSTAT

FAOSTAT_CommodityBalancesUnits

All units (such as "kg" or "tonnes") that may be used in commodity balances in FAOSTAT

FAOSTAT_ProducerPriceUnits

All units (such as "USD") that may be used in producer price data in FAOSTAT

3.1.7 _V6_Sets_FAOSTAT_LandUse

This file declares the sets where the FAOSTAT Land Use items are collected

In this code file, the following sets are defined:

FAOSTAT_LandUSeItemCode

All element codes that may be used in FAOSTAT

FAOSTAT_LandUseItems

All land use item names that may be used in FAOSTAT

FAOSTAT_LandUseItemCodeAndItems(FAOSTAT_LandUSeItemCode,FAOSTAT_LandUseItems)

Two dimensional set containing all land use item codes and names matched that may be used in FAOSTAT

FAOSTAT_LandUseElements

All land use elements (such as "Area") that are used in FAOSTAT

FAOSTAT_LandUseUnits

All land use units (such as "ha" or "million tonnes") that may be used in land use in FAOSTAT

3.1.8 _V6_Sets_FAOSTAT_Deforestation

This file declares the sets where the FAOSTAT Deforestation items are collected. See also section 3.15.

In this code file, the following sets are defined:

FAOSTAT_DeforestationItems

All deforestation item names ("Forest" and "Net Forest Conversion") that may be used in FAOSTAT

FAOSTAT_DeforestationElements

All deforestation elements (such as "Area") that are used in FAOSTAT

FAOSTAT_DeforestationUnits

All deforestation units (such as "ha" or "gigagrams") that may be used in FAOSTAT

3.1.9 _V6_Sets_FAOSTAT_OrganicSoils

This file declares the sets where the FAOSTAT Organic Soils items are collected. See also section 3.16.

In this code file, the following sets are defined:

FAOSTAT_OrganicSoilsItems

All organic soils item names (such as "cropland organic soils") that may be used in FAOSTAT

FAOSTAT_OrganicSoilsElements

All organic soils elements (such as "Area") that are used in FAOSTAT

FAOSTAT_OrganicSoilsUnits

All organic soils units (such as "ha" or "gigagrams") that may be used in FAOSTAT

3.1.10 _V6_Sets_ErbEtAl_Grasslands

This file declares the sets needed to read grassland data from (Erb, Gaube et al. 2007). For UNISECO, the grassland data is updated with the updated values from BioBaM.

In this code file, the following sets are defined:

ErbEtAl_GrasslandYieldClasses

Different grass yield classes

ErbEtAl_AverageYieldElement

Average grass yield elements such as tDM per ha

ErbEtAl_GrasslandNutrientElements

Grass nutrient elements such as tDM per ton or MJ per tDM

ErbEtAl_GrasslandDataCountriesAndRegions

Countries and regions as used in Erb et al. 2007 grassland data

ErbEtAl_GrasslandDataCountryList(ErbEtAl_GrasslandDataCountriesAndRegions)

Sub-set with countries as used in Erb et al. 2007 grassland data

ErbEtAl_GrasslandDataRegions(ErbEtAl_GrasslandDataCountriesAndRegions)

Sub-set with regions as used in Erb et al. 2007 grassland data

ErbEtAl_GrasslandDataCountriesInRegions(ErbEtAl_GrasslandDataRegions,ErbEtAl_GrasslandDataCountryList)

Two dimensional set containing all Erb et al. 2007 grassland regions matched to countries

3.1.11 _V6_Sets_FAOSTAT_Fertilizers

This file declares the sets that are needed to read FAOSTAT fertilizer data (items, elements and units)

In this code file, the following sets are defined:

FAOSTAT_FertilizerItems

Fertilizer items (such as "ammonium nitrate (AN)" etc.) used in FAOSTAT

FAOSTAT_FertilizerElements

Fertilizer elements (such as "Export quantity" etc.) used in FAOSTAT

FAOSTAT_FertilizerUnits

Fertilizer units (e.g. "tons") used in FAOSTAT

3.1.12 _V6_Sets_FAOSTAT_Population_HumanNutrReq

This file declares the sets where the FAOSTAT population data is collected and that are needed for human nutrient requirements.

In this code file, the following sets are defined:

FAOSTAT_PopulationItems

Population items (such as population number) used in FAOSTAT

FAOSTAT_PopulationElements

Population elements (such as "urban population") used in FAOSTAT

FAOSTAT_PopulationUnits

Population units ("1000 persons") used in FAOSTAT

3.1.13 _V6_Sets_VariousSources_HerdStructures

This file assigns the sets used to work with herd structures.

In this code file, the following sets are defined:

AnimalCategoriesInHerd_VariousSources

All animal categories according to age and reproductive and production status that may be used in the model - covers also some management aspects - not mutually exclusive

AnimalCategoriesInHerd_SOLmOLD(AnimalCategoriesInHerd_VariousSources)

All animal categories as used in older versions of SOLm (up to version V4)

CattleHerdCategories_SOLmOLD(AnimalCategoriesInHerd_VariousSources)

Subset with cattle herd categories as used in older versions of SOLm

DairyCattleHerdCategories_SOLmOLD(AnimalCategoriesInHerd_VariousSources)

Subset with DAIRY cattle herd categories as used in older versions of SOLm

BeefCattleHerdCategories_SOLmOLD(AnimalCategoriesInHerd_VariousSources)

Subset with BEEF cattle herd categories as used in older versions of SOLm

PigHerdCategories_SOLmOLD(AnimalCategoriesInHerd_VariousSources)
Subset with PIG herd categories as used in older versions of SOLm
AnimalCategoriesInHerd_FAOSTAT(AnimalCategoriesInHerd_VariousSources)
Subset with animal categories as used in FAOSTAT
AnimalCategoriesInHerd_IPCC2006(AnimalCategoriesInHerd_VariousSources)
Subset with animal categories as used in (IPCC 2006)
AnimalCategoriesInHerd_IPCC2006_Ruminants(AnimalCategoriesInHerd_VariousSources)
Subset with RUMINANT categories as used in (IPCC 2006)

3.1.14 _V6_Sets_GeneralModelSets_ForReadingData

This file declares some further sets that are needed to read in the data.

In this code file, the following sets are defined:
ProductionType
Captures the different types of production, such as organic or conventional, rain-fed or irrigated, etc. and also animal production systems - CURRENTLY only org/conv needed!

3.1.15 _V6_Sets_GeneralModelSets

This file declares all the sets needed in the model BESIDES the sets needed for filing in the data. All these sets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "GeneralModelSets".

In this code file, the following sets are defined:

3.1.15.1 Inputs

InputsCropsGrass
Captures the different inputs to the crop and grass activities, i.e. all mass and nutrient flows that are used by the activity; examples are seeds, fertilizer, land, etc.
InputsAnimals
Captures the different inputs to the animal activities, i.e. all mass and nutrient flows that are used by the activity; examples are feed, drinking water, etc.
InputsFishSeafood
Captures the different inputs to the fish and seafood activities, i.e. all mass and nutrient flows that are used by the activity; examples are feed, mangrove area, etc.
InputsForest
Captures the different inputs to the forest activities, i.e. all mass and nutrient flows that are used by the activity; examples are energy for logging, etc.
InputsOther
Captures the different inputs to the other activities, i.e. all mass and nutrient flows that are used by the activity

3.1.15.2 Outputs

OutputsCropsGrass
Captures the different outputs from the crop and grass activities, i.e. all mass and nutrient flows that result from an activity; thus, this covers e.g. MainOutput1 from Wheat (i.e. grains), Straw, Roots, etc., but also emissions, losses, etc.
YieldsCropsGrass(OutputsCropsGrass)
Captures the yields from the crop and grass activities - unit: biomass per ha

OutputsAnimals

Captures the different outputs from the animal activities, i.e. all mass and nutrient flows that result from an activity; thus, this covers e.g. meat, milk, manure, etc., but also emissions, losses, etc.

YieldsAnimals(OutputsAnimals)

Captures the different yields from the animal activities - units: biomass per head or APU

OutputsFishSeafood

Captures the different outputs from the fish and seafood activities, i.e. all mass and nutrient flows that result from an activity; thus, this covers e.g. Meat, but also emissions, losses, etc.

OutputsForest

Captures the different outputs from the forest activities, i.e. all mass and nutrient flows that result from an activity; thus, this covers e.g. wood, etc., but also emissions, losses, etc.

OutputsOther

Captures the different outputs from the other activities, i.e. all mass and nutrient flows that result from an activity

3.1.15.3 Other Characteristics of Activities

OtherCharCropsGrass

Characteristics of the crop or grass activity being undertaken that cannot be captured well by mass/nutrient flows; this can be the biodiversity loss or deforestation, but also monetary flows, etc.

OtherCharCropsGrass_PerHaBased(OtherCharCropsGrass)

OtherCharCropsGrass_PerTonYieldBased(OtherCharCropsGrass)

Other characteristics of the crop/grass activity that relate to per ha/per ton output

OtherCharAnimals

Characteristics of the animal activity being undertaken that cannot be captured well by mass/nutrient flows; this can be animal welfare aspects, but also monetary flows, etc.

OtherCharAnimals_PerHeadBased(OtherCharAnimals)

Other characteristics of the animal activity that relate to per head

OtherCharFishSeafood

Characteristics of the fish and seafood activity being undertaken that cannot be captured well by mass/nutrient flows; this can be seabed destruction, but also monetary flows, etc.

OtherCharForest

Characteristics of the forest activity being undertaken that cannot be captured well by mass/nutrient flows; this can be biodiversity loss, but also monetary flows, etc.

OtherCharOther

Characteristics of the other activity being undertaken that cannot be captured well by mass/nutrient flows

3.1.15.4 Animal types in herds

AnimalTypeInHerd

captures the different types of animals that are needed to build a full animal production unit, or a herd structure (i.e. with a dairy cow, there are several calves of different age, sires, etc.)

AnimalTypeInHerd_NoAggregates(AnimalTypeInHerd)

Subset of AnimalTypeInherd used to sum up to get the total amount of living animals - avoiding double counting by excluding AllAndAverageTypes, etc. - thus only relevant for animals with herd structure

CattleTypeInHerd(AnimalTypeInHerd)

Subset of AnimalTypeInHerd containing cattle types only

DairyCattleTypeInHerd(AnimalTypeInHerd)

Subset of AnimalTypeInHerd containing dairy cattle types only

BeefCattleTypeInHerd(AnimalTypeInHerd)

Subset of AnimalTypeInHerd containing beef cattle types only

PigTypeInHerd(AnimalTypeInHerd)

Subset of AnimalTypeInHerd containing pig types only

ProducingAnimals(AnimalTypeInHerd)

Subset of AnimalTypeInHerd containing producing animals only

SuckledAnimals(AnimalTypeInHerd)

Subset of AnimalTypeInHerd that is suckled

alias(AnimalTypeInHerd,AnimalTypeInHerd_2)

Duplicates the set AnimalTypeInHerd into an identical set AnimalTypeInHerd_2

MatchSucklingSuckledAnimals(AnimalTypeInHerdAnimalTypeInHerd_2)

Matching suckling and suckled AnimalTypeInHerd

3.1.15.5 Production systems

ProductionSystems

Captures the different systems of production, such as organic or conventional, rainfed or irrigated, etc. and also animal production systems

ProdSyst_OrgConAll(ProductionSystems)

Subset with the production systems “organic”, “conventional” and “All”

ProdSyst_OrgCon(ProductionSystems)

Subset with the production systems “organic” and “conventional”

ProdSyst_NoAggregates(ProductionSystems)

Subset with the production systems that are not aggregates of others (thus e.g. “All” is not included)

ProductionSystems_UsingMinNFert(ProductionSystems)

Production systems that use mineral fertilizers - to correctly adjust min fert use in the scenarios

3.1.15.6 Production conditions

ProductionConditions

Captures the production conditions from soil types and soil characteristics, climatic conditions, etc.

3.1.15.7 Commodities: nutrient contents and Other Characteristics

Contents

Captures nutrient contents of commodities

ContentsPerFreshMatterNutrients(Contents)

Captures non-food and non-feed nutrient contents of commodities (and also seeds) - only the per fresh matter values

ContentsPerFreshMatterFood(Contents)

Captures food nutrient contents of commodities (and also seeds) - only the per fresh matter values

ContentsPerFreshMatterFeed(Contents)

Captures feed nutrient contents of commodities (and also seeds) - only the per fresh matter values

CommodOtherChar

Captures other characteristics of commodities

3.1.15.8 Crop residues: nutrient contents, Other Characteristics and management

CropResContents

Captures nutrient contents of crop residues

CropResOtherChar

Captures other characteristics of crop residues

CropResManagement

Various parameters related to crop residue management - e.g. emissions, N losses, etc.

CropResManagement_NotSystemShares(CropResManagement)

Same as Set CropResManagement but without 'Quantity share in CropResMan system'

CropResManSystem

Crop residue management systems

CropResManSystemCropland(CropResManSystem)

all crop residue management systems from which crop residues can then be used on croplands (or grasslands as well) as it is centrally collected, or on croplands, as it is left there

CropResManSystemGrassland(CropResManSystem)

all crop residue management systems from which crop residues can then be used on grasslands

CropResManSystem_WithVolatInApplication(CropResManSystem)

all crop res man systems that show wolatilization in application (e.g. compost, biogas slurry,...)

3.1.15.9 Manure: nutrient contents, Other Characteristics and management

ManureContents

Captures nutrient contents of manure

ManureOtherChar

Captures other characteristics of manure

ManureManagement

Various parameters related to manure management - e.g. emissions, N losses, etc., UNITS - per t manure TS DM, if not indicated otherwise

ManureManSystem

Manure management systems

ManureManSystemCropland(ManureManSystem)

all manure management systems from which manure can then be used on croplands (or grasslands as well) as it is centrally collected

ManureManSystemGrassland(ManureManSystem)

all manure management systems from which manure can be used on grasslands only as it is left there

ManureManSystem_NoAverageNoTotal(ManureManSystem)

all manure management systems WITHOUT the average and total ones"

3.1.15.10 Fertilizer application characteristics

FertApplicCharact

Contains various characteristics of fertilizer application, such as N lost per ton N applied, etc.

3.1.15.11 Regions

Regions

Set containing all regions, countries, subregions (e.g. NUTS2 in the EU), etc. used in the model

alias(Regions,Regions_2)

Duplicates the set Regions into an identical set Regions_2

Countries(Regions)

Sub-set containing all countries (thus excludes more aggregate regions and sub-country level regions)

FAO_Africa(Regions)

FAO_Eastern_Africa(Regions)

FAO_Middle_Africa(Regions)

FAO_Northern_Africa(Regions)

FAO_Southern_Africa(Regions)

FAO_Western_Africa(Regions)
FAO_Americas(Regions)
FAO_Northern_America(Regions)
FAO_Central_America(Regions)
FAO_Caribbean(Regions)
FAO_South_America(Regions)
FAO_Asia(Regions)
FAO_Central_Asia(Regions)
FAO_Eastern_Asia(Regions)
FAO_Southern_Asia(Regions)
FAO_SouthEastern_Asia(Regions)
FAO_Western_Asia(Regions)
FAO_Europe(Regions)
FAO_Eastern_Europe(Regions)
FAO_Northern_Europe(Regions)
FAO_Southern_Europe(Regions)
FAO_Western_Europe(Regions)
FAO_Oceania(Regions)
FAO_Australia_NewZealand(Regions)
FAO_Melanesia(Regions)
FAO_Micronesia(Regions)
FAO_Polynesia(Regions)
FAO_EuropeanUnion(Regions)
SubSaharanAfrica(Regions)

The preceding sets define various sub-sets covering regions as used in FAOSTAT

FOFA2050_Rest_of_EAP(Regions)
FOFA2050_Rest_of_HIC(Regions)
FOFA2050_Rest_of_LAC(Regions)
FOFA2050_Rest_of_MNA(Regions)
FOFA2050_Rest_of_SAS(Regions)
FOFA2050_Rest_of_SSA(Regions)
FOFA2050_Rest_of_EU(Regions)
FOFA2050_Rest_of_ECA(Regions)

The preceding sets define various sub-sets covering regions as used in the data for the FAO 2050 projections from 2018, (FAO 2018)

Set Regions_Switzerland(Regions)
Set Regions_Austria(Regions)
Set Regions_Austria_Tal(Regions)
Set Regions_Austria_Huegel(Regions)
Set Regions_Austria_Berg(Regions)

The preceding sets define various sub-sets covering regions as used in the refined data for CH and AT used in the “Alpenprojekt”

3.1.15.12 Activities, sub-activities, aggregate activities

Activities

Set containing all activities used in the model - inclusive aggregates and sub-activities

alias(Activities,Activities_2)

Duplicates the set Activities into an identical set Activities _2

Livestock(Activities)

Subset containing all livestock activities

Livestock_NoAggregates(Activities)

Subset containing all livestock activities - and no aggregates such as "sheep and goats""

Ruminants(Activities)

Subset containing all ruminant activities

Poultry(Activities)

MonogastricsNonPoultry(Activities)

FishAndSeafood(Activities)

NonRuminants(Activities)

Livestock_NoCoupledOutputs(Activities)

Livestock_CoupledMeatMilk(Activities)

Livestock_CoupledMeatEggs(Activities)

Livestock_NonRuminantsWithHerdStructure(Activities)

Crops(Activities)

Subset containing all crop activities

Cereals(Activities)

Subset containing all cereal activities

Fruits(Activities)

Treenuts(Activities)

Pulses(Activities)

Legumes_NFixing(Activities)

Crops_NoNFixingLegumes(Activities)

OilCrops(Activities)

StarchyRoots(Activities)

SugarCrops(Activities)

Vegetables(Activities)

StimulantsSpices(Activities)

FibresRubber(Activities)

OtherCereals(Activities)

CitrusFruits(Activities)

Spices(Activities)

The preceding sets define sub-sets containing activity groups as indicated in their names

OtherVegetables(Activities)

Subset of all vegetables BESIDES Tomatoes

OtherOilcrops(Activities)

FibresNotCotton(Activities)

Fallows(Activities)

EnergyCrops(Activities)

The preceding sets define sub-sets containing activity groups as indicated in their names

ForageCrops(Activities)

Subset containing all forage crops

OtherFodderCrops(Activities)

Subset containing other fodder crops

GrassActivities(Activities)

Subset containing all grass activities

CoreGrassActivities(Activities)

Subset containing the key grass activities - no auxiliary ones for filing in data, etc.

CoreGrassActivitiesNoTEMPGrass(Activities)

Subset containing the key PERMANENT grass activities - no auxiliary ones for filing in data, etc.

TempMeadAndPastures(Activities)

Subset containing temporary meadows and pastures

TempAndPermMeadAndPastures(Activities)

Subset containing temporary and permanent meadows and pastures - thus covering all grasslands based on two categories from FAOSTAT

CropsAndCoreGrassActivities(Activities)

Subset of all agricultural area based activities, i.e. containing all Crops and GrassActivities

CropsAndTempGrassActivities(Activities)

Subset of all crop plus temporary grass activities

SingleCropGrassAndLivestockActivities(Activities)

Subset of single activities without aggregates - both crops and livestock

FOFA2050_SweetPotato_And_Yams(Activities)
FOFA2050_Rapeseed_And_Mustardseed(Activities)
FOFA2050_OtherCrops(Activities)
FOFA2050_OtherFibreCrops(Activities)
FOFA2050_OtherFruits(Activities)
FOFA2050_OtherOilseeds(Activities)
FOFA2050_OtherRootsAndTubers(Activities)
FOFA2050_OtherVegetables(Activities)
FOFA2050_CitrusFruits(Activities)
FOFA2050_DriedPulses(Activities)
FOFA2050_OtherCereals(Activities)

The preceding sets define various sub-sets covering activity groups as used in the data for the FAO 2050 projections from 2018, (FAO 2018)

OtherCrops(Activities)

3.1.15.13 Commodities

Commodities

Set containing all commodities used in the model - inclusive aggregates and sub-commodities

alias(Commodities,Commodities_2)

Duplicates the set Commodities into an identical set Commodities _2

ForageCommodities(Commodities)

Subset containing all forage commodities

Grasscommodities(Commodities)

Subset containing all grass commodities

ConcentrateCommodities(Commodities)

Subset containing all concentrate commodities

Commodities_SingleCommodities(Commodities)

Subset containing single commodities only - no additional aggregates; but some commodities may be listed twice, e.g. Pork and Pigmeat, etc., some may be aggregates from the original data (e.g. vegetables) - but usually only one of them then has an entry, I think.

Commodities_FeedGroups(Commodities)

Subset containing the aggregate feed groups (such as grass, concentrate, forage,…)

Commodities_FeedGroups_DetailedFeedingRations(Commodities)

Subset containing the aggregate feed groups (such as grass, concentrate, forage,…) as used for the DETAILED feeding rations calculations

Commodities_Feed(Commodities)

Set of the commodities that are used as feed

Commodities_Feed_Grass(Commodities)

Set of the GRASS commodities that are used as feed

Commodities_Feed_NoForageNoGrass(Commodities)

Set of the commodities that are used as feed WITHOUT forage and grass

Commodities_Feed_ForageCrops(Commodities)

Set of the FORAGE CROP commodities that are used as feed

Commodities_Feed_Cereals(Commodities)

Set of the CEREAL commodities that are used as feed

Commodities_Feed_OilCropsAndCakes(Commodities)

Set of the OIL CROP commodities that are used as feed

Commodities_Feed_Pulses(Commodities)

Set of the PULSES commodities that are used as feed

Commodities_Feed_Roots(Commodities)

Set of the ROOT commodities that are used as feed

Commodities_Feed_Sugar(Commodities)

Set of the SUGAR commodities that are used as feed

Commodities_Feed_OthersPlants(Commodities)

Set of the OTHER PLANT commodities that are used as feed

Commodities_Feed_OthersAnimals(Commodities)

Set of the OTHER ANIMAL commodities that are used as feed

CommoditiesDAQ_TradeFromToWorld(Commodities)

Commodity set used for the basic trade calculations as it includes indentical elements for all the DAQ categories and export/import - however ex and imp are to and from WORLD only

CommoditiesDAQ_TradeFromToWorldPrimary(Commodities)

Commodity set - only PRIMARY commodities that are produced in a country from an activity only, not from imported commodities - used for the basic trade calculations ...cf. explanations to the other sets above

CommoditiesDAQ_TradeFromToWorldPrimaryCrops(Commodities)

Commodity set - only PRIMARY CROP commodities that are produced in a country from an activity only, not from imported commodities - used for the basic trade calculations ...cf. explanations to the other sets above

Commodities_EquivalentToNonGrassFedAnimalActivityOutput(Commodities)

Commodities that are equivalent to the output of animal activities from animals WITHOUT grass in their feeding ration

CommoditiesDAQ_TradeFromToWorldPrimaryAnimals(Commodities)

Commodity set - only PRIMARY ANIMAL commodities that are produced in a country from an activity only, not from imported commodities - used for the basic trade calculations ...cf. explanations to the other sets above

CommoditiesDAQ_TradeFromToWorldPrimaryAnimals_GrassFed(Commodities)

Commodity set - only PRIMARY grass-fed ANIMAL commodities that are produced in a country from an activity only, not from imported commodities - used for the basic trade calculations ...cf. explanations to the other sets above

CommoditiesDAQ_TradeFromToWorldPrimaryAnimals_NonGrassFed(Commodities)

Commodity set - only PRIMARY non-grass-fed ANIMAL commodities that are produced in a country from an activity only, not from imported commodities - used for the basic trade calculations ...cf. explanations to the other sets above

CommoditiesDAQ_TradeFromToWorldDirectlyDerivedFromPrimary(Commodities)

Derived commodities that are directly derived from primary commodities for which trade and production data and thus DAQ are available

CommoditiesDAQ_TradeFTWorldDirDerivedFromPrimary_Animals(Commodities)

Derived ANIMAL commodities that are directly derived from primary commodities for which trade and production data and thus DAQ are available

CommoditiesDAQ_TradeFTWDirDerFromPrim_Animals_GrassFed(Commodities)

Derived grass-fed ANIMAL commodities that are directly derived from primary commodities for which trade and production data and thus DAQ are available

CommoditiesDAQ_TradeFTWDirDerFromPrim_Anim_NonGrassFed(Commodities)

Derived non-grass-fed ANIMAL commodities that are directly derived from primary commodities for which trade and production data and thus DAQ are available

CommoditiesDAQ_TradeFromToWorldCropCommodAndProducts(Commodities)

crop commodities that are pooled from primary commodities plus derived products

CommoditiesDAQ_TradeFromToWorld_FishSeafood(Commodities)

Commodity set - fish and seafood commodities - explanation cf. excel-file NutritiveFactors_CommoditySetCorrespondences

MatchCommAct_CommoditiesDAQ_TradeFromToWorld_CropsPrimary(Commodities,Activities,OutputsCropsGrass)

Matching set to match the crop commodities used for the basic trade calculations with the corresponding activities - used to derive the change in production of these commodities, if the production of the activities changes, for example

MatchCommAct_CommoditiesDAQ_TradeFromToWorld_CropsDerived(Commodities,Activities,OutputsCropsGrass)

Matching set to match the crop commodities used for the basic trade calculations with the corresponding activities - used to derive the change in production of these commodities, if the production of the activities changes, for example

MatchPrimDerivedCommodities_DAQ_TradeFromToWorld_CropsDerived(Commodities,Commodities_2)

Matching the derived commodities that are derived from primary commodities as captured in the previous set - DAQ and trade calculations for them are different than for the other derived commodities)

MatchPrimDerivedCommodities_DAQ_TradeFromToWorld_AnimalsDerived(Commodities,Commodities_2)

Matching the derived ANIMAL commodities that are derived from primary commodities as captured in the previous set - DAQ and trade calculations for them are different than for the other derived commodities)

MatchPrimDerivedCommodities_DAQ_TradeFTW_AnimalsDer_GrassFed(Commodities,Commodities_2)

Matching the derived grass-fed ANIMAL commodities that are derived from primary commodities as captured in the previous set - DAQ and trade calculations for them are different than for the other derived commodities)

MatchCommAct_CommodDAQ_TradeFTWorld_CropCommodAndProducts(Commodities,Activities,OutputsCropsGrass)

Matching set to match the crop commodities that are pooled from primary commodities plus derived products with the underlying activity

MatchCommAct_CommoditiesDAQ_TradeFromToWorld_AnimalsPrimary(Commodities,Activities,OutputsAnimals)

Matching set to match the animals commodities used for the basic trade calculations with the corresponding activities - used to derive the change in production of these commodities, if the production of the activities changes, for example

MatchCommAct_CommoditiesDAQ_TradeFTW_AnimalsPrimaryGrassFed(Commodities,Activities,OutputsAnimals)

Matching set to match the grass-fed animals commodities used for the basic trade calculations with the corresponding activities - used to derive the change in production of these commodities, if the production of the activities changes, for example

MatchCommAct_CommoditiesDAQ_TradeFTW_AnimPrimaryNonGrassFed(Commodities,Activities,OutputsAnimals)

Matching set to match the NON-grass-fed animals commodities used for the basic trade calculations with the corresponding activities - used to derive the change in production of these commodities, if the production of the activities changes, for example

MatchCommAct_CommoditiesDAQ_TradeFromToWorld_AnimalsDerived(Commodities,Activities,OutputsAnimals)

Matching set to match the animals commodities used for the basic trade calculations with the corresponding activities - used to derive the change in production of these commodities, if the production of the activities changes, for example

3.1.15.14 Some auxiliary sets

The following sets are also needed to read in data; in the code, they are structured as follows, here we do not add this structure between the files but display it at the beginning for a better overview:

14.1) Years
14.2) Temperatures
14.3) Greenhouse gases

the sets 14.1-14.3 are also needed to read in data, not only in the model runs

14.4) Sets related to mineral fertilizers
14.5) Sets related to population and human nutrition
14.6) Sets related to feeding rations
14.7) Some auxiliary matching sets for reading data (to be defined here and not further up, as they refer to the set Activities and Commodities)
14.8) Sets related to energy production

Years
Set that contains all the different years used in the model
BasisyearsOLD(Years)
these are the years /2005*2009/ used as basis years in older versions of SOLm, up to V4
Basisyears(Years)
these are the current default basis years /2009*2013/
BasisyearsSeed(Years)
these are the current default basis years to be used for averages of seed variables (shifted by one year, as they are used in the subsequent year only) /2008*2012/

Temperatures
Set of temperature values in degree Celsius - to be used for manure management emissions calculations, for example, etc.
Temperatures0to100Celsius(Temperatures)
generally used temperatures - range of 0 to 100 degree Celsius
TemperaturesBelow10(Temperatures)
TemperaturesAbove28(Temperatures)

GreenhouseGases
relevant greenhouse gases for agriculture, used to define the GWP and GTP, etc. in a flexible way as a parameter

MineralFertilizerType
Types of mineral fertilizers
MineralFertilizerProdTech
Mineral fertilizer production technologies
MinFertChar
Mineral fertilizer characteristics
PopulationGroups
population groups - all, male, female, children
Humans_InputsOutputsOtherCharacteristics
inputs to, outputs from and other characteristics of humans
FeedingRationOtherChar
Characteristics of feeding rations that is not captured in the set contents or so - e.g. quantity share in DM for all commodities, etc.

MatchFaostatLiveAnimalItems_Activities(FAOSTAT_LiveAnimalsItems,Activities)
Match_ActivityOutputsToCommodities_Crops(Activities,OutputsCropsGrass,Commodities)

Set to link the main outputs from the crop activities to commodities, that can then be further worked with on commodity level

Match_ActivityOutputsToCommodities_Animals(Activities,OutputsAnimals,Commodities)

Set to link the main outputs from the animal activities to commodities, that can then be further worked with on commodity level

Match_FeedCommoditiesToFeedCommodGroups(Commodities,Commodities_2)

Matches the commodities used as feed to the four main feed groups (Conc, Forage, Grass, Residues)

Match_FeedCommoditiesToFeedCommodGroups_MainByprodConc(Commodities,Commodities_2)

Matches the commodities used as feed to the main feed groups, separating main and byprod for concentrates (ConcMainProd, ConcByProd, Forage, Grass, Residues)

Match_FeedCommodToFeedCommodGroups_DetailedFeedRatio(Commodities,Commodities_2)

Matches the commodities used as feed to the main feed groups, using the groups used for the detailed feeding rations calculations

ConversionLevel

Primary or secondary energy

EnergyType

Type of energy (electricity heat etc.)

EnergySource

Energy source (biomass fossil etc.)

EnergyChar

Characteristics of the energy production (quantity or emissions etc.)

3.1.16 _V6_VariablesAndParameters

This file declares the key parameters and variables needed in SOLmV6

We first provide a detailed explanation of the structure:
the first basis of the model version V6 are activity units, i.e.

- the activities undertaken on a unit of agricultural land (hectare), with all the related inputs (e.g. seed), outputs (e.g. wheat grain) and other characteristics (e.g. risk to lead to deforestation)
- the activities undertaken to produce animal source food, framed in relation to one animal production unit, with all the related inputs (e.g. feed), outputs (e.g. milk) and other characteristics (e.g. risk to lead to antibiotic resistances)
- the activities undertaken to produce animal source food, framed in relation to SINGLE ANIMALS as constituents of an animal production unit, with all the related inputs, outputs and other characteristics
- the activities undertaken to produce fish and seafood, with "ton unprocessed fish/seafood biomass output" as the basic unit, , with all the related inputs (e.g. fuel energy), outputs (e.g. fish quantity) and other characteristics (e.g. seabed destruction)
- the activities undertaken on a unit of forest land (hectare), with all the related inputs (e.g. seedlings), outputs (e.g. wood) and other characteristics (e.g. biodiversity loss)
- any other activities - to be specified (could be used to capture insects, algae, vertical farming, cultured meat, etc.)

thereby,

- activities are any action that produces outputs from inputs, characterized by the above options: crops, animals, fish/seafood, forests, and other
- inputs are all mass/nutrient flows that are used when undertaking the activity
- outputs are all mass/nutrient flows that result from the activity
- other characteristics are all characteristics of the activity being undertaken that cannot be captured well by mass/nutrient flows; this can be the biodiversity loss or seabed destruction mentioned above, but also monetary flows, etc.

All these inputs, outputs and characteristics are displayed as parameters per unit activity, or, for animals, also per head.
Multiplication by the variable "quantity of activity units" then results in variables of input, output and characteristics quantities.

In this, we differentiate the sets inputs, outputs and other characteristics according to these fundamentally different activities as well (crops, animals, fish/seafood, forests, other).
But we keep one set activities, as we may later want to have agroforestry activities with crops AND animals included, etc.

The variables corresponding to totals values for per unit parameters (derived by multiplying the activity level times the per unit values) have the same names as the parameter, just with the letter "V" added at the beginning. Thus, the parameter is e.g. ActCropsGrass_Inputs, i.e. per hectare inputs, and the variable for the total is then VActCropsGrass_Inputs. All parameters and variables come with a number of dimensions, the first usually being Regions, the next Activities, then the specific aspects such as input indicators (e.g. ton nitrogen from mineral fertilizers per hectare), etc., then a dimension for ProductionSystems (such as organic or conventional) and ProductionConditions (currently not further differentiated, thus using the entry "All", but could capture aspects such as soil quality or climatic conditions).

Part of the outputs are then commodities (e.g. MainOutput1 from wheat is "Wheat grains" (or "Wheat") as commodities.
This is captured in the second basis of the model version V6, the commodity units:

These capture the specific outputs from the activities that are termed "commodities" and their nutrient contents and, where relevant, other characteristics (such as energy use for production, e.g. for Wheat flour from wheat grains), on a per ton basis.
Here, the differentiation between crops, animals, etc. is not used anymore.
All these contents and characteristics are displayed as parameters per ton commodity.
Multiplication by the variable "quantity of commodity" then results in variables of content and characteristics quantities.

Various commodities are related between each other via the commodity trees. This results in two further parameters:

- the Production shares - i.e. how much of wheat grain goes into wheat flour, how much into wheat beer, etc.; and
- the extraction rate, - i.e. how much wheat flour can be produced from one ton of wheat grain

The DETAILED TABLE OF CONTENTS for the parameter and variable definitions looks as follows:

2.3) Outputs from activities
2.4) Other characteristics of activities
2.5) Commodity quantities, nutrient contained and other characteristics
2.6) Commodity utilizations
2.7) Crop residue quantities, nutrient contained, other characteristics and management
2.8) Feeding rations quantities
2.9) Manure quantities, nutrients contained, other characteristics and management
2.10) Fertilizer application: nutrients and other characteristics
2.11) Import and export quantities
2.12) Commodities expressed in primary product equivalents

- 3) Auxiliary parameters
- 4) Auxiliary variables

Thus, in this code file, the following parameters and variables are defined:

3.1.16.1 Parameters

3.1.16.1.1 Activities: input parameters

ActCropsGrass_Inputs(Regions,Activities,InputsCropsGrass,ProductionSystems,ProductionConditions)

inputs to the crop and grass activities undertaken on a unit of agricultural land - UNIT: input per hectare

ActAnimalsAPU_Inputs(Regions,Activities,AnimalTypeInHerd,InputsAnimals,ProductionSystems,ProductionConditions)

inputs to the animal activities undertaken to produce animal source food - UNIT: input per Animal Production Unit

ActAnimalsHead_Inputs(Regions,Activities,AnimalTypeInHerd,InputsAnimals,ProductionSystems,ProductionConditions)

inputs to the animal activities undertaken to produce animal source food - UNIT: input per Animal Head

ActFishSeafood_Inputs(Regions,Activities,InputsFishSeafood,ProductionSystems,ProductionConditions)

inputs to the fish and seafood activities - UNIT: input per ton unprocessed fish or seafood biomass output

ActForest_Inputs(Regions,Activities,InputsForest,ProductionSystems,ProductionConditions)

inputs to the forest activities undertaken on a unit of forest land - UNIT: input per hectare

ActOthers_Inputs(Regions,Activities,InputsOther,ProductionSystems,ProductionConditions)

inputs to any other activities undertaken - UNIT: to be specified - default: input per ton unprocessed main biomass output

3.1.16.1.2 Activities: output parameters

ActCropsGrass_Outputs(Regions,Activities,OutputsCropsGrass,ProductionSystems,ProductionConditions)

outputs from the crop and grass activities undertaken on a unit of agricultural land - UNIT: output per hectare

ActAnimalsAPU_Outputs(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,ProductionSystems,ProductionConditions)

outputs from the animal activities undertaken to produce animal source food - UNIT: output per Animal Production Unit

ActAnimalsHead_Outputs(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,ProductionSystems,ProductionConditions)

outputs from the animal activities undertaken to produce animal source food - UNIT: output per Animal Head

ActFishSeafood_Outputs(Regions,Activities,OutputsFishSeafood,ProductionSystems,ProductionConditions)

outputs from the fish and seafood activities - UNIT: output per ton unprocessed fish or seafood biomass output

ActForest_Outputs(Regions,Activities,OutputsForest,ProductionSystems,ProductionConditions)

outputs from the forest activities undertaken on a unit of forest land - UNIT: output per hectare

ActOthers_Outputs(Regions,Activities,OutputsOther,ProductionSystems,ProductionConditions)

outputs from any other activities undertaken - UNIT: to be specified - default: output per ton unprocessed main biomass output

3.1.16.1.3 Activities: other characteristics

ActCropsGrass_OtherChar(Regions,Activities,OtherCharCropsGrass,ProductionSystems,ProductionConditions)

other characteristics of the crop and grass activities undertaken on a unit of agricultural land - UNIT: OtherChar per hectare or ton etc.

ActAnimalsAPU_OtherChar(Regions,Activities,AnimalTypeInHerd,OtherCharAnimals,ProductionSystems,ProductionConditions)

other characteristics of the animal activities undertaken to produce animal source food - UNIT: OtherChar per Animal Production Unit

ActAnimalsHead_OtherChar(Regions,Activities,AnimalTypeInHerd,OtherCharAnimals,ProductionSystems,ProductionConditions)

other characteristics of the animal activities undertaken to produce animal source food - UNIT: OtherChar per Animal Head

ActFishSeafood_OtherChar(Regions,Activities,OtherCharFishSeafood,ProductionSystems,ProductionConditions)

other characteristics of the fish and seafood activities - UNIT: OtherChar per ton unprocessed fish or seafood biomass output

ActForest_OtherChar(Regions,Activities,OtherCharForest,ProductionSystems,ProductionConditions)

other characteristics of the forest activities undertaken on a unit of forest land - UNIT: OtherChar per hectare or ton etc.

ActOthers_OtherChar(Regions,Activities,OtherCharOther,ProductionSystems,ProductionConditions)

other characteristics of any other activities undertaken - UNIT: to be specified - default: OtherChar per ton unprocessed main biomass output

3.1.16.1.4 Commodities: nutrient contents and other characteristics

Commod_Contents(Regions,Commodities,Contents,ProductionSystems,ProductionConditions)

nutrient contents of commodities - UNIT: units nutrient per ton commodity

Commod_OtherChar(Regions,Commodities,CommodOtherChar,ProductionSystems,ProductionConditions)

other characteristics of commodities - UNIT: OtherChar per ton commodity

3.1.16.1.5 Commodity tree parameters

Commod_ProductionShare(Regions,Commodities,Commodities_2,ProductionSystems)

production share of different commodities from the same higher level commodity in the commodity tree - UNIT: share (i.e. percentage divided by 100)

Commod_ExtractionRate(Regions,Commodities,Commodities_2,ProductionSystems)

extraction rate of a commodity from its higher level source commodity - UNIT: share (i.e. percentage divided by 100)

The following defines a parameter that captures how aggregate commodities are disaggregated into primary activities (e.g. "Bread" comes from "All Cereals" and those have to be disaggregated to "Wheat", "Rye", etc.). Assumptions on this are often very shaky and for now, much is determined by expert guess from Adrian Muller from August 2019 - or where sensible, it is allocated according to the relative shares of single commodities in the aggregate, if this information is available.

Commod_SingleInAggregateCommodShares(Regions,Commodities,Commodities_2,ProductionSystems)

Share of single commodities in aggregate commodities - UNIT: share (i.e. percentage divided by 100)

3.1.16.1.6 Crop residues: nutrient contents, other characteristics and management

CropResidues_Contents(Regions,Activities,OutputsCropsGrass,CropResContents,ProductionSystems,ProductionConditions)

nutrient contents of crop residues - UNIT: units nutrient per ton crop residues

CropResidues_OtherChar(Regions,Activities,OutputsCropsGrass,CropResOtherChar,ProductionSystems,ProductionConditions)

other characteristics of crop residues - UNIT: OtherChar per ton crop residues

CropResidues_Management(Regions,Activities,OutputsCropsGrass,CropResManagement,CropResManSystem,ProductionSystems,ProductionConditions)

values related to crop residues management - UNIT: units management values per ton crop residues

3.1.16.1.7 Feeding rations

FeedingRations_Contents(Regions,Activities,AnimalTypeInHerd,Commodities,Contents,ProductionSystems,ProductionConditions)

"nutrient contents of feed/Feeding rations - UNIT: units nutrient per ton feed"

FeedingRations_OtherChar(Regions,Activities,AnimalTypeInHerd,Commodities,FeedingRationOtherChar,ProductionSystems,ProductionConditions)

"other characteristics of feed/Feeding rations - UNIT: other characteristics per ton feed"

FeedingRationsHeads_Contents(Regions,Activities,AnimalTypeInHerd,Commodities,Contents,ProductionSystems,ProductionConditions)

nutrient contents of feed/Feeding rations - UNIT: units nutrient per animal head"

FeedingRationsHeads_OtherChar(Regions,Activities,AnimalTypeInHerd,Commodities,FeedingRationOtherChar,ProductionSystems,ProductionConditions)

"other characteristics of feed/Feeding rations - UNIT: other characteristics per animal head"

FeedingRationsAPU_Contents(Regions,Activities,AnimalTypeInHerd,Commodities,Contents,ProductionSystems,ProductionConditions)

"nutrient contents of feed/Feeding rations - UNIT: units nutrient per APU"

3.1.16.1.8 Manure: nutrient contents, other characteristics and management

Manure_Contents(Regions,Activities,AnimalTypeInHerd,ManureContents,ProductionSystems,ProductionConditions)

nutrient contents of manure - UNIT: units nutrient per ton manure

Manure_OtherChar(Regions,Activities,AnimalTypeInHerd,ManureOtherChar,ProductionSystems,ProductionConditions)

other characteristics of manure - UNIT: OtherChar per ton manure

Manure_Management(Regions,Activities,AnimalTypeInHerd,ManureManagement,ManureManSystem,Temperatures,ProductionSystems,ProductionConditions)

values related to manure management - UNIT: units management values per t manure TS DM if not indicated otherwise

3.1.16.1.9 Fertilizer application: nutrients and other characteristics

below: Activities_2 is Livestock plus also aggregates thereof, therefore the set Livestock is not enough and it needs to be activities

ManureApplication(Regions,Activities,Activities_2,AnimalTypeInHerd,FertApplicCharact,ProductionSystems,ProductionConditions)

Characteristics of manure application to activities; Livestock/AnimalTypeInHerd captures the source of manure - UNIT: Char per ton manure/nutrient

CropResAndBiomassApplication(Regions,Activities,Commodities,FertApplicCharact,ProductionSystems,ProductionConditions)

Characteristics of crop residue and other crop biomass application to activities; commodities captures the source of crop residue (other biomass - UNIT: Characteristics per ton crop res/nutrient

MinFertApplication(Regions,Activities,MineralFertilizerType,FertApplicCharact,ProductionSystems,ProductionConditions)

Characteristics of min fert application to activities; MineralFertilizerType captures the source of min fert - UNIT: Char per ton min fert/nutrient

3.1.16.1.10 Extraction rates

ExtractionRate_CommodityTree(Regions,Commodities)

Extraction rates of commodities from their source products (e.g. 0.75 for “wheat flour” from “wheat grains”) – UNIT: ratio (i.e. percentage/100)

3.1.16.2 Main variables

3.1.16.2.1 Amount of activity units

VActCropsGrass_QuantityActUnits(Regions,Activities,ProductionSystems,ProductionConditions)

total amount of activity units - UNIT: Number of hectares

VActAnimalsAPU_QuantityActUnits(Regions,Activities,AnimalTypeInHerd,ProductionSystems,ProductionConditions)

total amount of activity units - UNIT: Number of Animal Production Units (APUs)

VActAnimalsHead_QuantityActUnits(Regions,Activities,AnimalTypeInHerd,ProductionSystems,ProductionConditions)

total amount of activity units - UNIT: Number of animal heads

VActFishSeafood_QuantityActUnits(Regions,Activities,ProductionSystems,ProductionConditions)

total amount of activity units - UNIT: Tons of unprocessed fish or seafood

VActForest_QuantityActUnits(Regions,Activities,ProductionSystems,ProductionConditions)

total amount of activity units - UNIT: Number of hectares

VActOthers_QuantityActUnits(Regions,Activities,ProductionSystems,ProductionConditions) total amount of activity units - UNIT: to be specified - default: tons of unprocessed main biomass output

3.1.16.2.2 Inputs to activities

VActCropsGrass_Inputs(Regions,Activities,InputsCropsGrass,ProductionSystems,ProductionConditions)

total inputs to the crop and grass activities undertaken on agricultural land - UNIT: total input

VActAnimalsAPU_Inputs(Regions,Activities,AnimalTypeInHerd,InputsAnimals,ProductionSystems,ProductionConditions)

total inputs to the animal activities undertaken to produce animal source food - UNIT: total input

VActAnimalsHead_Inputs(Regions,Activities,AnimalTypeInHerd,InputsAnimals,ProductionSystems,ProductionConditions)

total inputs to the animal activities undertaken to produce animal source food - UNIT: total input

VActFishSeafood_Inputs(Regions,Activities,InputsFishSeafood,ProductionSystems,ProductionConditions)

total inputs to the fish and seafood activities - UNIT: total input

VActForest_Inputs(Regions,Activities,InputsForest,ProductionSystems,ProductionConditions)

total inputs to the forest activities undertaken on a unit of forest land - UNIT: total input

VActOthers_Inputs(Regions,Activities,InputsOther,ProductionSystems,ProductionConditions)

total inputs to any other activities undertaken - UNIT: total input

3.1.16.2.3 Outputs from activities

VActCropsGrass_Outputs(Regions,Activities,OutputsCropsGrass,ProductionSystems,ProductionConditions)

total outputs from the crop and grass activities undertaken on agricultural land - UNIT: total output

VActAnimalsAPU_Outputs(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,ProductionSystems,ProductionConditions)

total outputs from the animal activities undertaken to produce animal source food - UNIT: total output

VActAnimalsHead_Outputs(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,ProductionSystems,ProductionConditions)

total outputs from the animal activities undertaken to produce animal source food - UNIT: total output

VActFishSeafood_Outputs(Regions,Activities,OutputsFishSeafood,ProductionSystems,ProductionConditions)

total outputs from the fish and seafood activities - UNIT: total output

VActForest_Outputs(Regions,Activities,OutputsForest,ProductionSystems,ProductionConditions)

total outputs from the forest activities undertaken on a unit of forest land - UNIT: total output

VActOthers_Outputs(Regions,Activities,OutputsOther,ProductionSystems,ProductionConditions)

total outputs from any other activities undertaken - UNIT: total output

3.1.16.2.4 Other characteristics of activities

VActCropsGrass_OtherChar(Regions,Activities,OtherCharCropsGrass,ProductionSystems,ProductionConditions)

total other characteristics of the crop and grass activities undertaken on agricultural land - UNIT: total OtherChar

VActAnimalsAPU_OtherChar(Regions,Activities,AnimalTypeInHerd,OtherCharAnimals,ProductionSystems,ProductionConditions)

total other characteristics of the animal activities undertaken to produce animal source food - UNIT: total OtherChar

VActAnimalsHead_OtherChar(Regions,Activities,AnimalTypeInHerd,OtherCharAnimals,ProductionSystems,ProductionConditions)

total other characteristics of the animal activities undertaken to produce animal source food - UNIT: total OtherChar

VActFishSeafood_OtherChar(Regions,Activities,OtherCharFishSeafood,ProductionSystems,ProductionConditions)

total other characteristics of the fish and seafood activities - UNIT: total OtherChar

VActForest_OtherChar(Regions,Activities,OtherCharForest,ProductionSystems,ProductionConditions)

total other characteristics of the forest activities undertaken on a unit of forest land - UNIT: total OtherChar

VActOthers_OtherChar(Regions,Activities,OtherCharOther,ProductionSystems,ProductionConditions)

total other characteristics of any other activities undertaken - UNIT: total OtherChar

3.1.16.2.5 Commodity quantities, nutrient contained and other characteristics

VCommod_Quantity(Regions,Commodities,ProductionSystems,ProductionConditions)

total (domestically available) commodity quantity (DAQ) - UNIT: tons

VCommod_Contents(Regions,Commodities,Contents,ProductionSystems,ProductionConditions)

total nutrient contents of commodities - UNIT: total units nutrient

VCommod_OtherChar(Regions,Commodities,CommodOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of commodities - UNIT: total units OtherChar

3.1.16.2.6 Commodity utilizations

VCommod_Production(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity production - UNIT: tons

VCommod_StockChanges(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity stock changes - UNIT: tons

VCommod_Food(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity used for food - UNIT: tons

VCommod_Feed(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity used for feed - UNIT: tons

VCommod_Seed(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity used for seed - UNIT: tons

VCommod_Processing(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity used for processing - UNIT: tons

VCommod_Waste(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity lost or wasted - UNIT: tons

VCommod_Other(Regions,Commodities,ProductionSystems,ProductionConditions)

total quantity of commodity used for other uses - UNIT: tons

*for some, we also are interested in contents and other characteristics of these commodity utilizations:

VCommod_Food_Contents(Regions,Commodities,Contents,ProductionSystems,ProductionConditions)

total nutrient contents of commodity used for food - UNIT: total units nutrient

VCommod_Feed_Contents(Regions,Commodities,Contents,ProductionSystems,ProductionConditions)

total nutrient contents of commodity used for feed - UNIT: total units nutrient

VCommod_Waste_Contents(Regions,Commodities,Contents,ProductionSystems,ProductionConditions)

total nutrient contents of commodity lost or wasted - UNIT: total units nutrient

VCommod_Food_OtherChar(Regions,Commodities,CommodOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of commodity used for food - UNIT: total units OtherChar

VCommod_Feed_OtherChar(Regions,Commodities,CommodOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of commodity used for feed - UNIT: total units OtherChar

VCommod_Waste_OtherChar(Regions,Commodities,CommodOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of commodity lost or wasted - UNIT: total units OtherChar

3.1.16.2.7 Crop residue quantities, nutrient contained, other characteristics and management

VCropResidues_Quantity(Regions,Activities,OutputsCropsGrass,ProductionSystems,ProductionConditions)

total crop residue quantity - UNIT: tons

VCropResidues_Contents(Regions,Activities,OutputsCropsGrass,CropResContents,ProductionSystems,ProductionConditions)

total nutrient contents of crop residues - UNIT: total units nutrient

VCropResidues_OtherChar(Regions,Activities,OutputsCropsGrass,CropResOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of crop residues - UNIT: total units OtherChar

VCropResidues_Management(Regions,Activities,OutputsCropsGrass,CropResManagement,CropResManSystem,ProductionSystems,ProductionConditions)

total values related to crop residues management - UNIT: total units management values

3.1.16.2.8 Feeding rations quantities

VFeedingRations_Quantity(Regions,Activities,AnimalTypeInHerd,Commodities,ProductionSystems,ProductionConditions)

total quantity of feed - UNIT: tons

VFeedingRations_Contents(Regions,Activities,AnimalTypeInHerd,Commodities,Contents,ProductionSystems,ProductionConditions)

total nutrient contents of feed - UNIT: total units nutrient

VFeedingRations_OtherChar(Regions,Activities,AnimalTypeInHerd,Commodities,FeedingRationOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of feed/Feeding rations - UNIT: total other characteristics

3.1.16.2.9 Manure quantities, nutrients contained, other characteristics and management

VManure_Quantity(Regions,Activities,AnimalTypeInHerd,ProductionSystems,ProductionConditions)

total manure quantity - UNIT: tons

VManure_Contents(Regions,Activities,AnimalTypeInHerd,ManureContents,ProductionSystems,ProductionConditions)

total nutrient contents of manure - UNIT: total units nutrient

VManure_OtherChar(Regions,Activities,AnimalTypeInHerd,ManureOtherChar,ProductionSystems,ProductionConditions)

total other characteristics of manure - UNIT: total units OtherChar

VManure_Management(Regions,Activities,AnimalTypeInHerd,ManureManagement,ManureManSystem,Temperatures,ProductionSystems,ProductionConditions)

total values related to manure management - UNIT: total units management values

3.1.16.2.10 Fertilizer application: nutrients and other characteristics

below: Activities_2 is Livestock plus also aggregates thereof, therefore the set Livestock is not enough and it needs to be activities

VManureApplication(Regions,Activities,Activities_2,AnimalTypeInHerd,FertApplicCharact,ProductionSystems,ProductionConditions)

Total characteristics of manure application to activities; Livestock/AnimalTypeInHerd captures the source of manure - UNIT: total Char

VCropResAndBiomassApplication(Regions,Activities,Commodities,FertApplicCharact,ProductionSystems,ProductionConditions)

Total characteristics of crop residue and other crop biomass application to activities; commodities captures the source of crop residue/other biomass - UNIT: total Characteristic

VMinFertApplication(Regions,Activities,MineralFertilizerType,FertApplicCharact,ProductionSystems,ProductionConditions)

Total characteristics of min fert application to activities; MineralFertilizerType captures the source of min fert - UNIT: total Char

3.1.16.2.11 Import and export quantities

VImportQuantity(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions)

total commodity quantity IMPORTED into Regions FROM Regions_2 - UNIT: tons

VExportQuantity(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions)

total commodity quantity EXPORTED from Regions INTO Regions_2 - UNIT: tons

VImportLivingAnimalsHead(Regions,Regions_2,Activities,AnimalTypeInHerd,ProductionSystems,ProductionConditions)

total number of live animals IMPORTED into Regions FROM Regions_2 - UNIT: Number of animal heads

VExportLivingAnimalsHead(Regions,Regions_2,Activities,AnimalTypeInHerd,ProductionSystems,ProductionConditions)

total number of live animals EXPORTED from Regions INTO Regions_2 - UNIT: Number of animal heads

The following is for trade in beehives and other animal activities measured in APUs only:

VImportLivingAnimalsAPU(Regions,Regions_2,Activities,ProductionSystems,ProductionConditions)

total number of live animals IMPORTED into Regions FROM Regions_2 - UNIT: Number of Animal Production Units (APUs)

VExportLivingAnimalsAPU(Regions,Regions_2,Activities,ProductionSystems,ProductionConditions)

total number of live animals EXPORTED from Regions INTO Regions_2 - UNIT: Number of Animal Production Units (APUs)

VImportQuantity_Feed(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions)

total commodity FEED quantity IMPORTED into Regions FROM Regions_2 - UNIT: tons

VExportQuantity_Feed(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions)

total commodity FEED quantity EXPORTED into Regions FROM Regions_2 - UNIT: tons

3.1.16.2.12 Commodities expressed in primary product equivalents

The following variables are expressed in prim prod equivalents, derived from the corresponding variables that are reported in commodity quantities (cf. definitions above), by means of extraction rates, etc. There is no need to add activity and output dimensions, as this information is captured in the relevant matching sets as defined in "`_V6_ReadData_CommodityTrees_LinkActivitiesAndCommodities.gms`"

VPrimProd_Commod_Quantity(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT commodity quantity - UNIT: tons

VPrimProd_Commod_Production(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity production - UNIT: tons

VPrimProd_Commod_StockChanges(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity stock changes - UNIT: tons

VPrimProd_Commod_Food(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity used for food - UNIT: tons

VPrimProd_Commod_Feed(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity used for feed - UNIT: tons

VPrimProd_Commod_Seed(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity used for seed - UNIT: tons

VPrimProd_Commod_Processing(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity used for processing - UNIT: tons

VPrimProd_Commod_Waste(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity lost or wasted - UNIT: tons

VPrimProd_Commod_Other(Regions,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT quantity of commodity used for other uses - UNIT: tons

VPrimProd_ImportQuantity(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT commodity quantity IMPORTED into Regions FROM Regions_2 - UNIT: tons

VPrimProd_ExportQuantity(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT commodity quantity EXPORTED from Regions INTO Regions_2 - UNIT: tons

VPrimProd_Commod_Quantity_CropActivities(Regions,Commodities,Activities,OutputsCropsGrass,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT commodity quantity - linked to activities - UNIT: tons

VPrimProd_Commod_Quantity_AnimalActivities(Regions,Commodities,Activities,OutputsAnimals,ProductionSystems,ProductionConditions)

total PRIMARY PRODUCT EQUIVALENT commodity quantity - linked to activities - UNIT: tons

3.1.16.3 Auxiliary parameters

HumanCharacteristics(Regions,PopulationGroups,Humans_InputsOutputsOtherCharacteristics,Commodities)

any characteristics of humans - such as nutrient requirements, nutrient excretions and others - UNITS: characteristics/inputs/outputs per CAPITA. Thereby, commodities refer to the source of food, e.g. plant or animal based, but also cereals, oil crops, etc. or even finer on commodity level

SeedContents(Regions,Activities,Contents,ProductionSystems,ProductionConditions)

any characteristics of seeds - such as nutrient and DM contents, etc. - UNITS: per ton seed

MineralFertilizerCharacteristics(Regions,MineralFertilizerType,MineralFertilizerProdTech,MinFertChar,ProductionSystems)

any characteristics of mineral fertilizers - such as production emissions, etc. - UNITS: per ton min. fert. nutrient

GWP_GTP_SOLm(GreenhouseGases)

values for the radiative forcing of GHGs - GWP or GTP - to be set at the beginning and then to be used for all calculations

3.1.16.4 Auxiliary variables

VMineralFertilizerQuantity(Regions,MineralFertilizerType,MineralFertilizerProdTech,ProductionSystems)

total amount of mineral fertilizers from 'min. fert. production technology', applied on 'production systems' in 'region' - UNIT: tons N, P_2O_5, K_2O

VPopulationNumbers(Regions,PopulationGroups)

total population numbers per region - differentiated by population groups, such as male, female, children, etc. - UNIT: number of people

VEnergyProduction(Regions,ConversionLevel,EnergyType,EnergySource,EnergyChar)

total energy related variables - total production and supply, emissions, related CCS, etc. - UNIT: units total

3.1.17 _V6_ReadData_FAOSTAT_CropProduction

This code reads the following data from FAOSTAT:
VActCropsGrass_QuantityActUnits.l(Regions,Activities,"AllProdSyst","AllProdCond")
area harvested in hectares
VActCropsGrass_Outputs.l(Regions,Activities,"MainOutput1 (t)","AllProdSyst","AllProdCond")
main production in tonnes
VActCropsGrass_Inputs.l(Regions,Activities,"Seeds (t)","AllProdSyst","AllProdCond")
total seed inputs in tonnes
ActCropsGrass_Outputs(Regions,Activities,"MainOutput1 (t)","AllProdSyst","AllProdCond")
yield in tonnes per hectare

Then derive seed input per ha from seed quantity and areas harvested (better would be: area cropped, instead of harvested!). This is a very gross measure of seed use per ha to be improved, e.g. by standard seed application rates for crops, etc.
ActCropsGrass_Inputs(Regions,Activities,"Seeds (t)","AllProdSyst","AllProdCond")
seed input in tonnes per ha

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_CropProductionItems of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_CropProductionItems".

3.1.18 _V6_ReadData_FAOSTAT_ForageCropProduction

This code reads the following data from FAOSTAT:
VActCropsGrass_QuantityActUnits.l(Regions,Activities,"AllProdSyst","AllProdCond")
area harvested in hectares
VActCropsGrass_Outputs.l(Regions,Activities,"MainOutput1 (t)","AllProdSyst","AllProdCond")
main production in tonnes
ActCropsGrass_Outputs(Regions,Activities,"MainOutput1 (t)","AllProdSyst","AllProdCond")
yield in tonnes per hectare

Seed data is not available, but is assigned from similar crops by a specific matching filed defined in the code (e.g. sees for forage maize is taken to be the same as for maize, etc.) and from some specific data for certain crops as indicated in the code.
ActCropsGrass_Inputs(Regions,Activities,"Seeds (t)","AllProdSyst","AllProdCond")
seed input in tonnes per ha

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_ForageCropProductionItems of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_ForageCropProdItems".

IMPORTANT: This data has not been officially validated by FAOSTAT. FAOSTAT indicates that the data is not as reliable as the data made publicly available. This data is thus not anymore publicly available on the web, but it is provided on request. The data currently used in SOLm is from the original file "Production_Fodder crops 1990_2017_export SWS official 25.09.2018.xlsx" as sent to us per mail from FAO on 25.9.2018.

3.1.19 _V6_ReadData_FAOSTAT_LivestockProduction

This code reads the following data from FAOSTAT:
VActAnimalsHead_Outputs.l(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,"AllProdSyst","AllProdCond")

Production from animals in tonnes (derived from “number producing heads”*”yield per producing animal”)

VActAnimalsHead_QuantityActUnits.l(Regions,Activities,"Living","AllProdSyst","AllProdCond")

Number of animals (heads)

VActAnimalsAPU_Outputs.l(Regions,"Beehives",AnimalTypeInHerd,OutputsAnimals,"AllProdSyst","AllProdCond")

APU is animal production unit; Production from animals in tonnes (derived from “number of APUs”*”yield per APU”)

VActAnimalsAPU_QuantityActUnits.l(Regions,"Beehives",AnimalTypeInHerd,"AllProdSyst","AllProdCond")

Number of APUs

ActAnimalsHead_Outputs(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,"AllProdSyst","AllProdCond")

Yield per head producing animal in tonnes / head

ActAnimalsAPU_Outputs(Regions,"Beehives",AnimalTypeInHerd,OutputsAnimals,"AllProdSyst","AllProdCond")

Yield per APU in tonnes / APU

The APU data is used for “Beehives” only, all other animals are covered by the “head” data. Values are available for the subset FAOSTAT_Countries of the set Regions (displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_Countries”) and for the subset FAOSTAT_LiveAnimalsItems of the set Activities. The subset FAOSTAT_LiveAnimalsItems captures the set FAOSTAT_LivestockPrimaryItems in which the data is provided from FAOSTAT best on SOLmV6 Activities – in combination with output types, as described in the sets displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_LiveAnimalsItems”.

3.1.20 _V6_ReadData_FAOSTAT_Trade

This code reads the following data from FAOSTAT:

VImportQuantity(Regions,Regions_2,Commodities,"AllProdSyst","AllProdCond")

Import quantity of Commodities (from Regions_2 into Regions) in tonnes

VExportQuantity(Regions,Regions_2,Commodities,"AllProdSyst","AllProdCond")

Export quantity of Commodities (from Regions to Regions_2) in tonnes

VImportLivingAnimalsHead(Regions,"World",Activities,"Living","AllProdSyst","AllProdCond")

Import of living animals in heads, from “World” to Regions

VExportLivingAnimalsHead(Regions,"World",Activities,"Living","AllProdSyst","AllProdCond")

Export of living animals in heads, from Regions to “World”

VImportLivingAnimalsAPU(Regions,"World",”Beehives”,"AllProdSyst","AllProdCond")

Import of living animals in APU, from “World” to Regions

VExportLivingAnimalsAPU(Regions,"World",”Beehives”,"AllProdSyst","AllProdCond")

Export of living animals in APU, from Regions to “World”

The APU data is used for “Beehives” only, all other animals are covered by the “head” data. Values are available for the subset FAOSTAT_Countries of the set Regions, for the subset FAOSTAT_DetailedTradeMatrixItems of the set Commodities and for the subset FAOSTAT_TradeLiveAnimalsItems of the set Activities. These subsets are displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries” and “ActCommod_FAOSTATTrade”.

Importantly, the trade data for "China, Hong Kong SAR", "China, Macao SAR", "China, mainland" and "China, Taiwan Province of" has been summed to values for “China” and then these four parts have been dropped. Thus, the data is more consistent with the production data, etc., where only “China” is reported.

To assign correct sustainability impacts to the commodities available in a region, they have to be traced back to their countries of production (thus, e.g. Germany may report to import considerable quantities of soy from The Netherlands. This soy, however, is not grown in The Netherlands, but imported from other countries, say Brazil and the US. Thus, it is assumed that the soy imported into Germany from The Netherlands in fact originates from Brazil and the US. This information is not available in the FAO trade-data and additional calculations and assumptions are needed to derive such systematically for all trade flows to assign the direct linkage between country of production and final importing country. (Kastner, Kastner et al. 2011) provides procedures on how to do this, which are also implemented in SOLm. For further details on this, see section **Fehler! Verweisquelle konnte nicht gefunden werden.**.

3.1.21 _V6_ReadData_FAOSTAT_CommodityBalances

This code reads the following data from FAOSTAT:

VCommod_Quantity.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Domestically available quantity (DAQ) of Commodities in a Region in tonnes

VImportQuantity.l(Regions,"World",Commodities,"AllProdSyst","AllProdCond")

Imported quantity of Commodities from "World" into a Region in tonnes

VExportQuantity.l(Regions,"World",Commodities,"AllProdSyst","AllProdCond")

Exported quantity of Commodities to "World" from a Region in tonnes

VCommod_StockChanges.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Stock changes of Commodities in a Region in tonnes

VCommod_Production.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Production of Commodities in a Region in tonnes

VCommod_Feed.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Quantity of Commodities in a Region used as FEED in tonnes

VCommod_Seed.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Quantity of Commodities in a Region used as SEED in tonnes

VCommod_Food.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Quantity of Commodities in a Region used as FOOD in tonnes

VCommod_Waste.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Quantity of Commodities in a Region WASTED/LOST in tonnes

VCommod_Processing.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Quantity of Commodities in a Region used as PROCESSED FURTHER in tonnes

VCommod_Other.l(Regions,Commodities,"AllProdSyst","AllProdCond")

Quantity of Commodities in a Region used OTHERWISE (e.g. bioenergy) in tonnes

Commod_OtherChar(Regions,Commodities,"Util food (share)","AllProdSyst","AllProdCond")
Commod_OtherChar(Regions,Commodities,"Util feed (share)","AllProdSyst","AllProdCond")
Commod_OtherChar(Regions,Commodities,"Util seed (share)","AllProdSyst","AllProdCond")
Commod_OtherChar(Regions,Commodities,"Util waste (share)","AllProdSyst","AllProdCond")
Commod_OtherChar(Regions,Commodities,"Util other (share)","AllProdSyst","AllProdCond")
Commod_OtherChar(Regions,Commodities,"Util processing (share)","AllProdSyst","AllProdCond")

The parameters above capture the various utilization shares of commodities as described in their names; unit: share (derived by division of the respective utilization quantity by the total DAQ).

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subsets FAOSTAT_LivestockFishCommodityBalancesItems, FAOSTAT_CropCommodityBalancesItems, ForageCommodities of the set Commodities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "Commod_FAOCommodBalances".

3.1.22 _V6_ReadData_FAOSTAT_LandUse

This code reads the following data from FAOSTAT:

VActCropsGrass_QuantityActUnits.l(Regions,Activities,"AllProdSyst","AllProdCond")

Areas under various land use as captured in the corresponding Activities, in hectares

VActForest_QuantityActUnits.l(Regions,Activities,"AllProdSyst","AllProdCond")

Areas under various forest use as captured in the corresponding Activities, in hectares

Values are available for the subset FAOSTAT_Countries of the set Regions and for some elements of the subset FAOSTAT_LandUse of the set Activities as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_LandUse".

3.1.23 _V6_ReadData_FAOSTAT_Deforestation

This code reads the following data from FAOSTAT:
ActCropsGrass_OtherChar(Regions,Crops,"Deforestation (ha)","AllProdSyst","AllProdCond")
ActCropsGrass_OtherChar(Regions,GrassActivities,"Deforestation (ha)","AllProdSyst","AllProdCond")

unit: ha deforestation per ha agricultural area, positive - if no deforestation occurs, this value is zero, as this is already coded like this in the FAO data

ActCropsGrass_OtherChar(Regions,Crops,"Deforest GHG emissions (tCO2e)","AllProdSyst","AllProdCond")
ActCropsGrass_OtherChar(Regions,GrassActivities,"Deforest GHG emissions (tCO2e)","AllProdSyst","AllProdCond")

unit: t CO_2eq emissions from deforestation per ha agricultural area, positive - if no deforestation occurs, this value is zero, as this is already coded like this in the FAO data

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subsets Crops and GrassActivities of the set Activities as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries” and “SOLmV6_CropsGrassAct”. For further details, see section 3.15.

3.1.24 _V6_ReadData_FAOSTAT_OrganicSoils

This code reads the following data from FAOSTAT:
ActCropsGrass_OtherChar(Regions,Crops,"CultOrgSoils (ha)","AllProdSyst","AllProdCond")
ActCropsGrass_OtherChar(Regions,GrassActivities,"CultOrgSoils (ha)","AllProdSyst","AllProdCond")

unit: ha managed organic soils per ha agricultural area, positive - if no organic soils are used, this value is zero, as this is already coded like this in the FAO data

ActCropsGrass_OtherChar(Regions,Crops,"CultOrgSoils GHG emissions (tCO2e)","AllProdSyst","AllProdCond")
ActCropsGrass_OtherChar(Regions,GrassActivities,"CultOrgSoils GHG emissions (tCO2e)","AllProdSyst","AllProdCond")

unit: t CO2eq emissions from managed organic soils per ha agricultural area, positive - if no organic soils are managed, this value is zero, as this is already coded like this in the FAO data

ActCropsGrass_OtherChar(Regions,Crops,"CultOrgSoils C emissions (tCO2e)","AllProdSyst","AllProdCond")
ActCropsGrass_OtherChar(Regions,GrassActivities,"CultOrgSoils C emissions (tCO2e)","AllProdSyst","AllProdCond")

unit: t C emissions in tCO2eq from managed organic soils per ha agricultural area, positive - if no organic soils are managed, this value is zero, as this is already coded like this in the FAO data

ActCropsGrass_OtherChar(Regions,Crops,"CultOrgSoils N2O emissions (tCO2e)","AllProdSyst","AllProdCond")
ActCropsGrass_OtherChar(Regions,GrassActivities,"CultOrgSoils N2O emissions (tCO2e)","AllProdSyst","AllProdCond")

unit: t N2O emissions in tCO2eq from managed organic soils per ha agricultural area, positive - if no organic soils are managed, this value is zero, as this is already coded like this in the FAO data

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subsets Crops and GrassActivities of the set Activities as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries” and “SOLmV6_CropsGrassAct”. For further details, see section 3.16.

3.1.25 _V6_ReadData_ErbEtAl_Grasslands

This code reads the following data from (Erb, Gaube et al. 2007):
ActCropsGrass_Outputs(Regions,Activities,"MainOutput1 (t DM)","AllProdSyst","AllProdCond")

Grassland DM yield, tDM/ha

ActCropsGrass_Outputs(Regions,Activities,"MainOutput1 (t)","AllProdSyst","AllProdCond")

Grassland yield, t/ha

Commod_Contents(Regions,Activities,"FeedME in DM (MJ)","AllProdSyst","AllProdCond")

Grass FeedME contents in DM (MJ/tDM)

Commod_Contents(Regions,Activities,"FeedXP in DM (t)","AllProdSyst","AllProdCond")

Grass FeedXP contents in DM (tXP/tDM)

Commod_Contents(Regions,Activities,"DM (t)","AllProdSyst","AllProdCond")

Grass DM contents (tDM/t)

Commod_Contents(Regions,Activities,"FeedME (MJ)","AllProdSyst","AllProdCond")

Grass FeedME contents (MJ/t)
Commod_Contents(Regions,Activities,"FeedXP (t)","AllProdSyst","AllProdCond")
Grass FeedXP contents (tXP/t)
Commod_Contents(Regions,Activities,"N (t)","AllProdSyst","AllProdCond")
Grass N contents (tN/t)
Commod_Contents(Regions,Activities,"FeedGE (MJ)","AllProdSyst","AllProdCond")
Grass FeedGE contents (MJ/t)
Commod_Contents(Countries,Activities,"P2O5 (t)","AllProdSyst","AllProdCond")
Grass P contents (tP2O5/t)
VActCropsGrass_Outputs.l(Regions,Activities,"MainOutput1 (t)","AllProdSyst","AllProdCond")
Total grass production quantity in tonnes
VCommod_Quantity.l(Regions,Activities,"AllProdSyst","AllProdCond")
Total grass domestically available quantity in tonnes (usually equals production – at least in the baseline FAO data, there is no trade in grass)
VCommod_Production.l(Regions,Activities,"AllProdSyst","AllProdCond")
Total grass commodity production quantity in tonnes (equals VActCropsGrass_Outputs.l)
VCommod_Feed.l(Regions,Activities,"AllProdSyst","AllProdCond")
Total grass domestically available quantity used as feed in tonnes (usually equals DAQ, i.e. 100% is used as feed)

Values are available for the subset ErbEtAl_GrasslandDataCountryList of the subset FAOSTAT_Countries of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" (ErbEtAl_GrasslandDataCountryList is not displayed there but can be found in the code; it differs by absence of some small countries only), and for the elements "Permanent meadows and pastures" and "Temporary meadows and pastures" of the set Activities, displayed in the set GrassActivities in "SOLmV6_CropsGrassAct".

3.1.26 _V6_ReadData_FAOSTAT_Fertilizers

This code reads the following data from FAOSTAT:
VMineralFertilizerQuantity.l(Regions,"mineral N fert (N)","AllMinFertProdTech","AllProdSyst")
VMineralFertilizerQuantity.l(Regions,"mineral P fert (P2O5)","AllMinFertProdTech","AllProdSyst")
VMineralFertilizerQuantity.l(Regions,"mineral K fert (K2O)","AllMinFertProdTech","AllProdSyst")
This is mineral fertilizer quantities used in Regions, units are tons nutrients: N, P2O5, K2O.

Values are available for the subset FAOSTAT_Countries of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries".

3.1.27 _V6_ReadData_FAOSTAT_WOSY_DetailedFBS

*This is an old code module using FAOSTAT data on detailed food balance sheets that is not publicly available. It played a role in in SOLmV2; currently not in use and not updated to SOLmV6 as the publicly available data is good enough as a default for what is needed here; will likely be deleted soon.

3.1.28 _V6_ReadData_FAOSTAT_Population

This code reads the following data from FAOSTAT:
VPopulationNumbers.l(Regions,PopulationGroups)
Number of people in the respective population groups ("PopulationAll", "Male", "Female")

Values are available for the subset FAOSTAT_Countries of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries".

3.1.29 _V6_ReadData_VariousSources_HumanNutrientRequirements

This code reads the following data:

HumanCharacteristics(Regions,"PopulationAll",Humans_InputsOutputsOtherCharacteristics,"All commodities")

This contains the average human nutrient requirements (kcal ADER (average daily energy requirements), kcal MDER (minimum daily energy requirements), gram protein and gram fat, all on a per capita and per day basis. Could be differentiated per commodity groups, but currently, it is the total. Sources: (Walpole, Prieto-Merino et al. 2012) or (FAO WFP and IFAD 2012) – this latter data was provided by FAOSTAT, it does not seem to be publicly available – to be checked again).

Values are available for the subset FAOSTAT_Countries of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries".

3.1.30 _V6_ReadData_VariousSources_CropGrassNutrientRequirementsData

Currently, this code does not read any data; in earlier versions of SOLm, up to V4, it has been used to read some data, but the quality of it was low. In the code, the nutrient requirement has then be changed to exclusively be a parameter that is derived rather than taken from some data base. Thus, the nutrient requirement is currently derived from nutrient output in produced biomass, etc. – if better data is available, this can be read and used in the scenario definitions to overwrite the default derived from output nutrient quantities.

3.1.31 _V6_ReadData_VariousSources_MainOutputNutrientContentsData

This code reads the following data from a number of sources – for details, see the code file:

Commod_Contents(Countries,Commodities,Contents,"AllProdSyst","AllProdCond")

Contents is N, P2O5, DM, partly ME, GE, XP contents in fresh and partly also in dry matter, i.e. values per ton commodity or per ton DM commodity

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries". Commodities covers the primary output commodities only, i.e. it is FAOSTAT_CropProductionItems, forage crops in NON_FAOSTAT_CropProductionItems, and livestock primary products - a selection from livestock commodities that most closely capture the main outputs from livestock activities, all as described in in the sheet "CommodNutrientContents" in the excel-file "SOLm_Documentation_Appendix.xlsx".

3.1.32 _V6_ReadData_VariousSources_ResidueSharesAndNutrientContentsData

This code reads the following data from a number of sources – for details, see the code file:

CropResidues_Contents(Regions,Activities,"Average residues (t)",CropResContents,"AllProdSyst","AllProdCond")

Residues N-, P2O5-, DM-contents, before management, unit: tN, tP2O5, tDM/t

CropResidues_OtherChar(Regions,Activities,"Average residues (t)",CropResOtherChar,"AllProdSyst","AllProdCond")

Residue shares, i.e. residues in DM per DM main output: "Residue share t DM / t DM MainOutput1"

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet "FAOSTAT_CropProductionItems" and the elements "Temporary meadows and pastures" and "Permanent meadows and pastures" of the set Activities, displayed in the set GrassActivities in the sheet "SOLmV6_CropsGrassAct".

3.1.33 _V6_ReadData_VariousSources_SeedCharacteristicsData

This code reads the following data from a number of sources – for details, see the code file:

ActCropsGrass_Inputs(Regions,Activities,"Seeds (t)","AllProdSyst","AllProdCond")

Seed inputs in ton seeds per ha

SeedContents(Regions,Activities,Contents,"AllProdSyst","AllProdCond")

Seed N, P2O5 and DM contents in tons per ton seed

The sets Regions (basically: all countries) contains the elements as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "Regions_FAOCropProd" and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet "FAOSTAT_CropProductionItems".

3.1.34 _V6_ReadData_FAOSTAT_ProducerPrices

This code reads the following data from FAOSTAT:

Commod_OtherChar(Regions,Commodities,"Producer price ($)","AllProdSyst","AllProdCond")

Producer prices in $ per ton commodity

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Commodities contains the elements from the set FAOSTAT_ProducerPriceItems as displayed in the sheet "Commod_FAOProducerPrices".

3.1.35 _V6_ReadData_IPCC_GWP_GTPData

This code reads the following data from IPCC:

GWP_GTP_SOLm(GreenhouseGases)

Global warming/temperature potential for the three key greenhouse gases in agriculture, CO2, CH4, N2O.

3.1.36 _V6_ReadData_LuEtAl_NDepositionData

This code reads the following data from (Lu, Jiang et al. 2013):

ActCropsGrass_Inputs(Regions,Activities,"N deposition (tN)","AllProdSyst","AllProdCond")

ActForest_Inputs(Regions,"Forest","N deposition (tN)","AllProdSyst","AllProdCond")

Atmospheric N deposition in tN per hectare

The data is world-sub-region average values from the reference named above. Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Activities are the elements from the subset Crops as displayed in the sheet "Activities_NDeposition_LuEtAl", as well as the elements "Forest" and "Temporary meadows and pastures" and "Permanent meadows and pastures" from the set Activities.

3.1.37 _V6_ReadData_VariousSources_NFixationData

This code reads the following data from various sources (mainly from (Herridge, Peoples et al. 2008)), for details see the code file:

ActCropsGrass_OtherChar(Regions,Activities,"N fixation per ton MainOutput1 (tN)","AllProdSyst","AllProdCond")

N fixation per ton main output, tons N fixed per ton main output.

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet

“FAOSTAT_CropProductionItems” and the set NON_FAOSTAT_CropProductionItems as displayed in sheet “CommodNutrientContents”.

3.1.38 _V6_ReadData_VariousSources_SoilErosionData

This code reads the following data from various sources (for details, see the code file and section **Fehler! Verweisquelle konnte nicht gefunden werden.** or (Schader, Muller et al. 2015)):

ActCropsGrass_OtherChar(Regions,Activities,"Soil water erosion (t soil lost)","AllProdSyst","AllProdCond")

Water erosion of soil, tons soil lost per hectare.

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_Countries” and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet “FAOSTAT_CropProductionItems” and the elements "Temporary meadows and pastures" and "Permanent meadows and pastures" of the set Activities, displayed in the set GrassActivities in the sheet “SOLmV6_CropsGrassAct”, and forage and other forage crops in NON_FAOSTAT_CropProductionItems in the sheet “CommodNutrientContents”.

3.1.39 _V6_ReadData_VariousSources_IrrigationWaterData

This code reads the following data from various sources, mainly AQUASTAT and Water Footprint Network (for details, see the code file and section 3.17):

ActCropsGrass_Inputs(Regions,Activities,"Irrigation water – irrigated areas (m3)","AllProdSyst","AllProdCond")

Irrigation water use per hectare irrigated crop.

ActCropsGrass_Inputs(Regions,Activities,"Irrigation water – averaged over all areas (m3)","AllProdSyst","AllProdCond")

Irrigation water use per hectare average crop: Irrigation water use averaged over all cropping areas, i.e. each hectare cropped needs on average this amount of irrigation water (which reflects the share of areas irrigated and the irrigation water use per irrigated area).

ActCropsGrass_OtherChar(Regions,Activities,"Share irrigated areas (share)","AllProdSyst","AllProdCond")

Share of area per crop that is irrigated, unit: share.

ActCropsGrass_OtherChar(Regions,Activities,"Water scarcity indicator (index)","AllProdSyst","AllProdCond")

A region-specific index for water scarcity; captures the pressure on water sources from irrigation.

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_Countries” and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet “FAOSTAT_CropProductionItems” and the elements "Temporary meadows and pastures" and "Permanent meadows and pastures" of the set Activities, displayed in the set GrassActivities in the sheet “SOLmV6_CropsGrassAct”, and forage and other forage crops in NON_FAOSTAT_CropProductionItems in the sheet “CommodNutrientContents”.

YET TO BE COMPLETED

3.1.40 _V6_ReadData_VariousSources_AnimalWelfareData

This code reads the following data from various sources (for details, see the code file and section 3.18):

ActAnimalsHead_OtherChar(Regions,Activities,"AllAndAverageTypes","General animal welfare index",
"AllProdSyst","AllProdCond")
ActAnimalsHead_Inputs(Regions,"Cattle",CattleTypeInHerd, "General animal welfare index","AllProdSyst","AllProdCond")
ActAnimalsHead_Inputs(Regions,"Pigs",PigTypeInHerd, "General animal welfare index","AllProdSyst","AllProdCond")

General animal welfare index per animal head and year.

ActAnimalsHead_OtherChar(Regions,Activities,"AllAndAverageTypes","General parasite infestation index",
"AllProdSyst","AllProdCond")

ActAnimalsHead_Inputs(Regions,"Cattle",CattleTypeInHerd, "Genera parasite infestation index","AllProdSyst","AllProdCond")
ActAnimalsHead_Inputs(Regions,"Pigs",PigTypeInHerd, "General parasite infestation index","AllProdSyst","AllProdCond")

General parasite infestation index per animal head and year.

ActAnimalsHead_Inputs(Regions,"Cattle",”Producing_Dairy_Cattle”, "Mastitis incidence index","AllProdSyst","AllProdCond")

Mastitis incidence index per producing dairy cattle head and year.

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element “Game” of the set Activities. These subsets are displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries” and “FAOSTAT_LiveAnimalsItems”. The cattle/PigTypeInHerd values are identical to the averages for all cattle/pig.

YET TO BE COMPLETED

3.1.41 _V6_ReadData_VariousSources_PesticidesData

This code reads the following data from various sources (for details, see the code file and section 3.19 or (Schader, Muller et al. 2015)):

ActCropsGrass_OtherChar(Regions,Activities,"Aggreg. Pest. use level (index)","AllProdSyst","AllProdCond")

This is an aggregate pest use level index (dimensionless) – given on a per hectare basis; it is derived from indices on ease of access to pesticides per country, level of legislation on pesticides per country and pesticide use intensity of crops.

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_CropProductionItems of the set Activities. These subsets are displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries” and “FAOSTAT_CropProductionItems”.

3.1.42 _V6_ReadData_IPCC2006_RiceCroppingEmissionsData

This code reads the following data from (IPCC 2006) (for details, see the code file):

ActCropsGrass_Outputs(Countries,"Rice, paddy","CH4 flooded rice (t CH4)","AllProdSyst","AllProdCond")

CH4 emissions of rice, t CH4 per hectare

Values are available for the subset FAOSTAT_Countries of the set Regions. This subset is displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries”

3.1.43 _V6_ReadData_VariousSources_HerdStructures

This code reads the following data on the herd structures as derived in (Schader, Muller et al. 2015), see also section 3.6:

VActAnimalsHead_QuantityActUnits(Regions,"Cattle",AnimalTypeInHerd,"AllProdSyst","AllProdCond")
VActAnimalsHead_QuantityActUnits(Regions,"Pigs",AnimalTypeInHerd,"AllProdSyst","AllProdCond")

Number of animals in each of the herd structure types (such as “Sows”, “Boars”, “Producing_Dairy_Cattle”, etc.) .

Values are available for the subset FAOSTAT_Countries of the set Regions and for the elements “Cattle” and “Pigs” of the set Activities. This subset is displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_Countries”. The set AnimalTypeInHerd covers the core herd animal types as used in SOLm. For all other livestock activities than “Cattle” and “Pigs”, there is no herd structure and the element used there is thus “AllAndAverageTypes”. For “Cattle” and “Pigs”, the elements are part of the set AnimalTypeInHerd as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “SOLm_HerdStructure”.

This data on the herd structures is calculated separately based on the values for the FAOSTAT living and producing animals by means of a maximum entropy model. For details see section 3.6 and (Schader, Muller et al. 2015). The numbers for each herd type category are then set in relation to these two values (living/Producing), and those shares are then applied to scenario values of living/producing animals to get herd structures in the scenarios, thus assuming that the herd structure does not change. Currently, these herd structures are derived for cattle and pigs only.

This is the generic data to get some general herd structure for the FAOSTAT animal numbers. The quality varies strongly between countries, and wherever possible better data should be used (as e.g. for the EU from Eurostat, cf. section/file add new data animal numbers).

3.1.44 _V6_ReadData_VariousSources_AnimalProductionUnits

This file does not yet contain any code or data – to be designed yet. The aim would be to more systematically capture the animal production unit (APU) view on animal production, cf. the description towards the end of section 1.3.

3.1.45 _V6_ReadData_VariousSources_AnimalLiveweightData

This code reads the following data from various sources, mainly (IPCC 2006):
ActAnimalsHead_OtherChar(Regions,Activities,AnimalTypeInHerd,"Liveweight (t)","AllProdSyst","AllProdCond")
Live weight per animal head, in tons.

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_LiveAnimalsItems". The set AnimalTypeInHerd covers the core herd animal types as used in SOLm as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "SOLm_HerdStructure".

3.1.46 _V6_ReadData_VariousSources_AnimalDrinkingWaterRequirementData

This code reads the following data from various sources, for details see the code file:
ActAnimalsHead_OtherChar(Regions,Activities,"AllAndAverageTypes","Drinking water (m3)","AllProdSyst","AllProdCond")
ActAnimalsHead_Inputs(Regions,"Cattle",CattleTypeInHerd,"Drinking water (m3)","AllProdSyst","AllProdCond")
ActAnimalsHead_Inputs(Regions,"Pigs",PigTypeInHerd,"Drinking water (m3)","AllProdSyst","AllProdCond")
Drinking water requirement per animal head and year, in m^3

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element "Game" of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_LiveAnimalsItems". The cattle/PigTypeInHerd values are identical to the averages for all cattle/pig.

3.1.47 _V6_ReadData_VariousSources_FeedingRationsData

This code reads the following data from (Herrero, Havlik et al. 2013):
ActAnimalsHead_OtherChar(Regions,Activities,"AllAndAverageTypes",OtherCharAnimals,"AllProdSyst","AllProdCond")
Quantity share of aggregate feed groups in total DM feed uptake

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element "Game" of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and

“FAOSTAT_LiveAnimalsItems”. OtherCharAnimals is the share in feed on the level of aggregate feed groups, i.e. it covers the elements "Concentrates in Feed DM (share)", "Forage crops in Feed DM (share)", "Grass in Feed DM (share)" and "Residues in Feed DM (share)".

The same values are also reported in another parameter, with a focus on commodities supplied as feed rather than on animals and what they take in as feed:

FeedingRations_OtherChar(Regions,Activities,AnimalTypeInHerd,Commodities,"Quantity share in DM (share)","AllProdSyst","AllProdCond")
FeedingRations_OtherChar(Regions,"Cattle",CattleTypeInHerd,Commodities,"Quantity share in DM (share)","AllProdSyst","AllProdCond")
FeedingRations_OtherChar(Regions,"Pigs",PigTypeInHerd,Commodities,"Quantity share in DM (share)","AllProdSyst","AllProdCond")

Quantity share of aggregate feed groups in total DM feed uptake

Regions and Activities are as above, Commodities cover aggregate feed groups, i.e. the elements “AggregateFeedConcentrates_Commodity”,“AggregateFeedForageCrops_Commodity”,“AggregateFeedGrass_Commodity”,“AggregateFeedResidues_Commodity”. The set AnimalTypeInHerd covers the core herd animal types as used in SOLm as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “SOLm_HerdStructure”.

Currently, SOLm works with more detailed feeding rations on commodity level (cf. XXX) and the feeding rations based on the four broad feed groups are not used anymore in the calculations (cf. XXX).

3.1.48 _V6_ReadData_VariousSources_AnimalNutrientRequirementsData

This code reads the following data from various sources, for details see the code file:

ActAnimalsHead_OtherChar(Regions,Livestock,AnimalTypeInHerd,"FeedME_req_Total (MJ)","AllProdSyst","AllProdCond")

Total per animal head requirements of feed metabolisable energy ME in MJ/head/year

ActAnimalsHead_OtherChar(Regions,Livestock,AnimalTypeInHerd,"FeedXP_req_Total (t)","AllProdSyst","AllProdCond")

Total per animal head requirements of feed crude protein XP in tons/head/year

ActAnimalsHead_OtherChar(Regions,Livestock,AnimalTypeInHerd,"FeedGE_req_Total (MJ)","AllProdSyst","AllProdCond")

Total per animal head requirements of feed gross energy GE in MJ/head/year

ActAnimalsHead_OtherChar(Regions,"Cattle","Producing_Dairy_Cattle",OtherCharAnimals,"AllProdSyst","AllProdCond")

Additional parameters used to derive the ME requirements for cattle, related to energy use for maintenance, walking, pregnancy and milking

ActAnimalsHead_OtherChar(Regions,Livestock,AnimalTypeInHerd,"XPperME_InFeedReq (gXP/MJ)","AllProdSyst","AllProdCond")

An auxiliary parameter to derive protein contents XP values from metabolisable energy ME values, in gXP per MJ ME

ActAnimalsHead_OtherChar(Regions,Livestock,"AllAndAverageTypes","UE_per_GE (share)","AllProdSyst","AllProdCond")

Urinary energy expressed as fraction of GE

ActAnimalsHead_OtherChar(Regions,Livestock,"AllAndAverageTypes","Animal specific FeedGE cont (MJ/t)","AllProdSyst","AllProdCond")

Default value for feed GE contents in MJ/ton

ActAnimalsHead_OtherChar(Regions,Livestock,"AllAndAverageTypes","Digestibility of Feed (%)","AllProdSyst","AllProdCond")

Digestibility of feed as a percentage

Commod_Contents(Regions,"Milk, Whole","Milk solid contents (t)","AllProdSyst","AllProdCond")

Milk solid contents in tons per ton milk

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element “Game” of the set Activities. – which is the same as the SOLmV6-subset Livestock of the set Activities. These subsets are displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheets “FAOSTAT_Countries” and “FAOSTAT_LiveAnimalsItems”. OtherCharAnimals is specifically for Dairy cattle, specific parameters to derive ME requirements: "FeedME_Req_MilkProd (MJ/head)", "FeedME_Req_Maintenance (MJ/head)", "FeedME_Req_Walking (MJ/head)", "FeedME_Req_Pregnancy (MJ/head)". The set AnimalTypeInHerd covers the core herd animal types as used in SOLm as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “SOLm_HerdStructure”.

3.1.49 _V6_ReadData_VariousSources_EntericFermentationEmissionsData

This code reads the following data from (IPCC 2006), for details see the code file:
FeedingRationsHeads_OtherChar(Countries,Livestock,AnimalTypeInHerd,Commodities,"Percentage GE in feed converted to enteric CH4","AllProdSyst","AllProdCond")

Percentage GE in feed converted to enteric CH4, used to derive the enteric fermentation emissions later in the model (in Steering 2).

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element "Game" of the set Activities. – which is the same as the SOLmV6-subset Livestock of the set Activities (for enteric fermentation, data is provided for Ruminants, Pigs and Chickens only) These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_LiveAnimalsItems". The set AnimalTypeInHerd covers the core herd animal types for Cattle, Pigs and "AllAndAverageTypes" for other ruminants and chickens as used in SOLm as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "SOLm_HerdStructure". Commodities cover aggregate feed groups, i.e. the elements "AggregateFeedConcentrates_Commodity","AggregateFeedForageCrops_Commodity","AggregateFeedGrass_Commodity","AggregateFeedResidues_Commodity".

3.1.50 _V6_ReadData_VariousSources_CropResidueManagementData

This code reads the following data, mainly from (IPCC 2006), for details see the code file:
CropResidues_Management(Regions,Activities,"Average Residues (t)",CropResManagement,CropResManSystem,ProductionSystems,"AllProdCond")

Various aspects of crop residue management, such as crop residue management system shares, nutrients lost, emissions, and nutrients contained in the output from the crop residue management systems, available for field application, etc., see below.

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet "FAOSTAT_CropProductionItems" and the elements "Temporary meadows and pastures" and "Permanent meadows and pastures" of the set Activities, displayed in the set GrassActivities in the sheet "SOLmV6_CropsGrassAct". CropResManSystem covers the crop residue management systems as displayed in the sheet "CropResManagement" in the excel-file "SOLm_Documentation_Appendix.xlsx" ProductionSystems are "AllProdSyst" and for the element "Quantity share in CropResMan system" in CropResManagement also "Convent" and "Organic".

There are values for the following elements of the set CropResManagement (which set is also displayed in sheet CropResManagement in the excel-file "SOLm_Documentation_Appendix.xlsx"):

- "Quantity share in CropResMan system"
- "Crop res man CH4 (tCH4)"
- "Crop res man N2O (tN2O)"
- "Crop res man N loss (tN)"
- "Crop res N for areas (tN)"
- "Crop res man share P lost (tP2O5/tP2O5 in crop res)"
- "Crop res man P loss (tP2O5)"
- "Crop res P for areas (tP2O5)"

If not indicated otherwise, the units are per ton crop residues.

3.1.51 _V6_ReadData_VariousSources_ManureExcretionData

This code reads the following data, mainly from (IPCC 2006), for details see the code file:
ActAnimalsHead_OtherChar(Regions,Livestock,AnimalTypeInHerd,"N in manure per ton liveweight (tN/t lw/y)","AllProdSyst","AllProdCond")

t N excreted in manure per ton liveweight and year

ActAnimalsHead_OtherChar(Regions,Livestock,AnimalTypeInHerd,"P in manure per ton liveweight (tP2O5/t lw/y)","AllProdSyst","AllProdCond")

t P2O5 excreted in manure per ton liveweight and year

Manure_OtherChar(Countries,Livestock,AnimalTypeInHerd,"Ash content in feed DM (share)","AllProdSyst","AllProdCond")

Share ash in feed dry matter (used to derive volatile and total solids in manure)

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element "Game" of the set Activities. – which is the same as the SOLmV6-subset Livestock of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_LiveAnimalsItems". The set AnimalTypeInHerd covers the core herd animal types as used in SOLm as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "SOLm_HerdStructure".

3.1.52 _V6_ReadData_VariousSources_ManureManagementData

This code reads the following data, mainly from (IPCC 2006), for details see the code file:

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Quantity share in ManureMan system" ,ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

Share of total manure quantity in the different manure management systems

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"MCF: CH4 conversion factor (%)" ,ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

Methane conversion factor (%) – used in some IPCC calculations on CH_4 emissions from manure management.

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Bo: max. CH4 prod. cap. (m3CH4/kgVS)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

Methane production capacity per quantity volatile solids (VS) in manure; another parameter used for deriving CH_4 emissions from manure management $m^3CH_4/kgVS$

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Manure man N volat (% of N in manure)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

N volatized from manure during manure management (% of total N in manure)

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Manure man N leach (% of N in manure)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

N leached from manure during manure management (% of total N in manure)

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Manure man N2O dir (tN2O-N/tN)" ,ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

Direct N_2O emissions from manure management (tN_2O-N emitted per tN in manure)

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Manure man N2O-N from N volat (tN/tN volat)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

N_2O from N volatized during manure management: ton N per ton N volatized

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Manure man N2O-N from N leach (tN/tN leach)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

N_2O from N leached during manure management: ton N per ton N leached

Manure_Management(Regions,Livestock,AnimalTypeInHerd,"Manure man P loss as % of P in manure (%)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

P lost from manure during manure management (% of total P in manure)

Values are available for the subset FAOSTAT_Countries of the set Regions and for the subset FAOSTAT_LiveAnimalsItems plus the element "Game" of the set Activities – which is the same as the SOLmV6-subset Livestock of the set Activities. These subsets are displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheets "FAOSTAT_Countries" and "FAOSTAT_LiveAnimalsItems". The set AnimalTypeInHerd covers the core herd animal types as used in SOLm as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "SOLm_HerdStructure".

3.1.53 _V6_ReadData_VariousSources_MineralFertilizerProductionEmissionsData

This code reads the following data from (Wood and Cowie 2004), for details see the code file:
MineralFertilizerCharacteristics(Countries,"mineral N fert (tN)","AllMinFertProdTech","t CO2e/tN production","AllProdSyst")
MineralFertilizerCharacteristics(Countries,"mineral P fert (tP2O5)","AllMinFertProdTech","t CO2e/tP2O5 production","AllProdSyst")

Greenhouse gas emissions from mineral N and P fertilizer production: tCO_2e/ton N or P_2O_5

Values are available for the subset FAOSTAT_Countries of the set Regions. This subset is displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_Countries”.

3.1.54 _V6_ReadData_VariousSources_FertilizerApplicationData

This code reads the following data from (IPCC 2006), for details see the code file:
ManureApplication(Countries,Activities,Livestock,"AllAndAverageTypes",FertApplicCharact,"AllProdSyst","AllProdCond")

Various emission characteristics (see below) related to application of manure from source livestock to crops and grassland (“AllAndAverageTypes” is AnimalTypeInHerd)

CropResAndBiomassApplication(Countries,Activities,"All Residues",FertApplicCharact,"AllProdSyst","AllProdCond")

Various emission characteristics (see below) related to application of crop residues and biomass to crops and grassland

MinFertApplication(Countries,Activities,"mineral N fert (N)",FertApplicCharact,"AllProdSyst","AllProdCond")

Various emission characteristics (see below) related to application of mineral fertilizers to crops and grassland

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_Countries” and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet “FAOSTAT_CropProductionItems” and the subset CoreGrassActivities of the set Activities, displayed in the sheet “SOLmV6_CropsGrassAct”. The SOLmV6-subset Livestock of the set Activities is displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “FAOSTAT_LiveAnimalsItems”.

There are values for the following elements of FertApplicCharact:

- "N2O-N per kg fertilizer N applied (tN/tN)"
- "Volatized N as percentage of fertilizer N applied (%)"
- "Leached N as percentage of fertilizer N applied (%)"
- "N2O-N per kg N volatized from fert applic (share)"
- "N2O-N per kg N leached from fert applic (share)"

3.1.55 _V6_ReadData_VariousSources_NH3Emissions

This code reads the following data from various sources, for details see the code file:
MinFertApplication(Countries,Activities,"mineral N fert (N)","NH3-N as percentage of fertilizer N applied (%)","AllProdSyst","AllProdCond")

N lost as NH3 from mineral fertilizer N application; unit: NH3-N as percentage of fertilizer N applied (%)

ManureApplication(Countries,Activities,Livestock,"AllAndAverageTypes","NH3-N as percentage of fertilizer N applied (%)","AllProdSyst","AllProdCond")

N lost as NH3 from manure N application (source: Livestock); unit: NH3-N as percentage of manure N applied (%)

CropResAndBiomassApplication(Countries,Activities,"All Residues","NH3-N as percentage of fertilizer N applied (%)","AllProdSyst","AllProdCond")

N lost as NH3 from crop residue and other biomass N application; unit: NH3-N as percentage of crop residues and other biomass N applied (%)

Manure_Management(Countries,Livestock,AnimalTypeInHerd,"Manure man NH3-N (% of N in manure)",ManureManSystem,"AllAndAverageTemp","AllProdSyst","AllProdCond")

N lost as NH3 from manure N during manure management; unit: NH3-N as percentage of manure N in the management system (%)

The NH_3 emissions are PART OF the N-volatilization already addressed in a previous code file. They are NOT ADDITIONAL to this. Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet "FAOSTAT_CropProductionItems" and the subset CoreGrassActivities of the set Activities, displayed in the sheet "SOLmV6_CropsGrassAct". The SOLmV6-subset Livestock of the set Activities is displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_LiveAnimalsItems". The set AnimalTypeInHerd covers the core herd animal types as used in SOLm as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "SOLm_HerdStructure".

3.1.56 _V6_ReadData_VariousSources_OrganicYieldGapsData

This code reads the following data from various sources, for details see the code file and section 2.3.2:
ActCropsGrass_OtherChar(Regions,Activities,"organic yield gap (ratio org/conv yield)","AllProdSyst","AllProdCond")
ActAnimalsHead_OtherChar(Regions,Livestock,ProducingAnimals,"organic yield gap (ratio org/conv yield)","AllProdSyst","AllProdCond")
Organic yield gap for crop, grass and livestock activities: ratio organic/conventional

Values are available for the subset FAOSTAT_Countries (which is Countries in SOLmV6) of the set Regions as displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_Countries" and Activities contains the elements from the set FAOSTAT_CropProductionItems as displayed in the sheet "FAOSTAT_CropProductionItems" and the subset NON_FAOSTAT_CropProductionItems (which also covers some GRASS activities!) as displayed in sheet "CommodNutrientContents". The SOLmV6-subset Livestock of the set Activities is displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "FAOSTAT_LiveAnimalsItems".

3.1.57 __SOLmV5_DataDerivedBaseline_DetailedFeedingRations

This module is currently not operational for the baseline in SOLmV6. It is about more detailed – commodity-specific – feeding rations.

Currently, this is not executed and default values for detailed feeding rations are assigned in section 4.4. of the code module `_V6_BaselineValues_ForModelRuns` (cf. section 3.2.6).

3.1.58 _V6_ReadData_VariousSources_CED

This code reads the following data from various sources, for details see the code file:
CED is "Cumulative Energy Demand", a Live-Cycle-Assessment (LCA)-based measure for non-renewable energy use. This data is read in considerable detail for various crop and livestock management related processes and aggregated in SOLm to the following two indicators, available for all crops, grass and livestock activities: "Total CED (MJ)" and "Total GWP from CED (tCO2e)". The CED data is taken from ecoinvent 2.0 ((Nemecek, Heil et al. 2007)) and converted to GWP values by means of process-specific conversion factors (see code file), also taken from ecoinvent 2.0.

Check and improve this data – e.g. with the farm model!

3.1.59 _V6_ReadData_VariousSources_FishAndSeafoodData

This code reads data on fish and seafood production systems – yet to be written

3.1.60 _V6_ReadAdditionalData_SwitzerlandAustria

This code reads new more detailed baseline data for Switzerland and Austria, as used in a completed project on the alpine region. See section 3.5.1 for details.

This code is currently not used.

3.1.61 _V6_ReadAdditionalData_NUTS2_EU

This code reads new more detailed baseline data for NUTS2 data for the EU as used in the H2020 project UNISECO. See section 3.5.2 for details.

This code has yet to be comleted.

3.1.62 _V6_ReadAdditionalData_AnimalNumbers

This reads better data on animal numbers (incl. herd structure) for the EU-28, based on Eurostat. See section 3.5.3 for details.

3.1.63 _V6_ReadAdditionalData_GHGInventories

This section reads various parameter and variable values as used in the national GHG inventories and replaces the default values with them. Currently, this is done for Austria and Switzerland and then allows a detailed consistency check with the national GHG inventory values. See section 3.5.4 for details.

3.1.64 _V6_ReadAdditionalData_Prices

This reads better data for commodity prices for Switzerland, as used in some calculations for the NFP69 project on sustainable and healthy diets. See section 3.5.5 for details.

3.1.65 _V6_ReadAdditionalData_FeedData

This file contains better data on feed quantities, currently for feed imports to Switzerland, as used in the project for zhaw/greenpeace. See section 3.5.6 for details.

3.1.66 _V6_DataDerivedBaseline_SomeHerdStructureParameters

This file derives the relation of the various animal types in the herd to the total living animal number for the baseline and for the animals with herd structure in the baseline: cattle and pigs. This is then the basis to derive the herd structure for the total animal numbers in scenarios, etc.
Thus, the following data is provided:

ActAnimalsHead_OtherChar(Regions,"Cattle",AnimalTypeInHerd,"Share animal type in total living animals",ProductionSystems,ProductionConditions)
ActAnimalsHead_OtherChar(Regions,"Pigs",AnimalTypeInHerd,"Share animal type in total living animals",ProductionSystems,ProductionConditions)

> Shares of animal type in herd in relation to all living animals, for cattle and pigs (as only those have herd structures in the baseline)

These parameters are thus available for the same elements for Regions etc. as the source data, i.e. the variables on animal numbers for animal types in herds, cf. section 3.1.43.

3.1.67 _V6_ReadData_CommodityTrees_LinkActivitiesAndCommodities

This file contains the code for linking the outputs from Activities to the elements of Commodities. This is done by means of the commodity trees, extraction rates and shares of processing lines from one parent commodity. The basis for the default values for this is the FAO document "Technical Conversion Factors for Agricultural Commodities" from 1996 (there is no newer encompassing information available from FAO; for specific countries and commodities, better information can be searched for and added as new data, as described at the beginning of section 3.1 and in section 3.5). It is organized in seven parts referring to different types of commodities (such that are aggregates of other, such with well-defined co-products, such that are equivalent to primary outputs from activities, etc.). It defines the corresponding sets, the parameters “ExtractionRates” and the primary product equivalent values of commodities, etc. – a detailed description is provided in section 3.13. The seven sets are also listed and described shortly in section 3.2.1.

3.1.68 _V6_VariablesAndParameters_ModelRun

This file declares the parameters and variables for the model runs. In this, it is identical to the file “`_V6_VariablesAndParameters.gms`” described in section 3.1.16 and defines largely the same parameters and variables besides adding a scenario-dimension to each of these parameters and variables, amending their names with a suffix “_MR” for “model run” and defining the corresponding scenario set. The general procedure to define the model-run-parameters and –variables is thus as follows: add “_MR” (for model run) to the parameter/variable names and add the scenario dimension at the end of them.

The set Scenarios is displayed in the excel-file “SOLm_Documentation_Appendix.xlsx” in the sheet “Scenarios”.

After having defined the model run parameters and variables, those for which baseline values are available are assigned so, i.e. so that the model run parameter and variables with scenario dimension “Baseline” are set equal to the baseline values as read in the previous files.

The file also defines some additional parameters and variables and also leaves some away, first, in an additional section:
- 0) Define scenario set
    This defines the Set Scenarios /…/;

And then on several places:
In 1.7) FeedingRations, the following is NOT defined (but it is then defined in the Var and param file for steering_2: section 3.2.4, _V6_VariablesAndParameters_ModelRun_ForSteeringFile2)
FeedingRationsHeads_Contents_MR(Regions,Activities,AnimalTypeInHerd,Commodities,Contents,ProductionSystems,ProductionConditions,Scenarios)
    nutrient contents of feed/Feeding rations - UNIT: units nutrient per animal head"
In 4) Auxiliary variables the following is added:
AUX_Inverse_VActCropsGrass_Outputs_MR(Regions,Activities,OutputsCropsGrass,ProductionSystems,ProductionConditions,Scenarios)
    an auxiliary variable - the INVERSE of the total outputs from the crop and grass activities undertaken on agricultural land - UNIT: 1 divided by total output

Furthermore, it adds a whole section for the assignment of the baseline values:
- 5) Initialise par/var for model runs with baseline values
    5.1) Main parameters
    5.2) Main variables

3.1.69 _V6_ReadData_FAOSTAT_FOFA2050

This code reads the data for the food system projections to 2050 as presented in (FAO 2018). See section 3.5.7 for details.

3.1.70 _V6_ReadData_VariousSources_BioenergySR15

This code reads the data for the food system projections to 2050 and 2100 as presented in (IPCC 2018). See section 3.5.8 for details.

3.1.71 _V6_StreamlineInitialData

This file streamlines the initial data as produced by the SteeringFile1 and then collected in the following gdx-files (they are generated in the code file "`_V6_OutputFiles_SteeringFile1.gms`", see section 3.1.72):

- 'GeneralModelSets'
- 'GeneralModelParameters_Inputs'
- 'GeneralModelParameters_Outputs'
- 'GeneralModelParameters_OtherChar'
- 'GeneralModelParameters_Various'
- 'GeneralModelParameters_Auxiliary'
- 'GeneralModelVariables_ActivityQuantities'
- 'GeneralModelVariables_Inputs'
- 'GeneralModelVariables_Outputs'
- 'GeneralModelVariables_OtherChar'
- 'GeneralModelVariables_Various'
- 'GeneralModelVariables_Trade'
- 'GeneralModelVariables_CommodityTree'
- 'GeneralModelVariables_Auxiliary'
- 'FOFA2050_BioeSR15_Etc_Data_InModelRunEntities'

Going through these files, several useless, unimportant or confusing assignments have been identified. These are dropped and partly replaced further down in this code file. And several missing assignments have been identified and have been added. This file is then run before the gdx-files are read out in the code file "`_V6_OutputFiles_SteeringFile1.gms`" for further use in SteeringFile2.

In a later version of SOLm, one could incorporate all this directly in the preceding code files or change those such as to avoid the missing/useless assignments corrected here.

3.1.72 _V6_OutputFiles_SteeringFile1

This file contains the code to produce the following gdx-files. Subsequently it is displayed in detail what is contained in each of those files.

- GeneralModelSets.gdx
- GeneralModelParameters_Inputs.gdx
- GeneralModelParameters_Outputs.gdx
- GeneralModelParameters_OtherChar.gdx
- GeneralModelParameters_Various.gdx
- GeneralModelParameters_Auxiliary.gdx
- GeneralModelVariables_ActivityQuantities.gdx
- GeneralModelVariables_Inputs.gdx
- GeneralModelVariables_Outputs.gdx
- GeneralModelVariables_OtherChar.gdx
- GeneralModelVariables_Various.gdx
- GeneralModelVariables_Trade.gdx
- GeneralModelVariables_CommodityTree.gdx
- GeneralModelVariables_Auxiliary.gdx
- FOFA2050_BioeSR15_Etc_Data_InModelRunEntities.gdx

Detailed contents: below, the code directly from the GAMS-file is displayed. For each gdx-file to be generated, it has an "execute_unload" statement that defines the gdx-file to be generated: the statement is followed by the name of the new file and after this, all sets, parameters and variables to be included in the file are listed, ending with a semicolon ";". After this comes a comment section between "$ontext$" and "$offtext$" in detail presenting what is contained in each parameter and variable in the gdx-files.

All these sets, parameters and variables are then available to be read by subsequent code, currently the second steering file where the model runs are executed.

3.1.72.1 GeneralModelSets.gdx'

```
execute_unload 'GeneralModelSets'
InputsCropsGrass
InputsAnimals
InputsFishSeafood
InputsForest
InputsOther
OutputsCropsGrass
YieldsCropsGrass
OutputsAnimals
YieldsAnimals
OutputsFishSeafood
OutputsForest
OutputsOther
OtherCharCropsGrass
OtherCharCropsGrass_PerHaBased
OtherCharCropsGrass_PerTonYieldBased
OtherCharAnimals
OtherCharAnimals_PerHeadBased
OtherCharFishSeafood
OtherCharForest
OtherCharOther
AnimalTypeInHerd
AnimalTypeInHerd_NoAggregates
CattleTypeInHerd
DairyCattleTypeInHerd
BeefCattleTypeInHerd
PigTypeInHerd
ProducingAnimals
SuckledAnimals
AnimalTypeInHerd_2
MatchSucklingSuckledAnimals

ProductionSystems
ProdSyst_OrgConAll
ProdSyst_OrgCon
ProdSyst_NoAggregates
ProductionSystems_UsingMinNFert
ProductionConditions
Contents
ContentsPerFreshMatterNutrients
ContentsPerFreshMatterFood
ContentsPerFreshMatterFeed
CommodOtherChar
ManureOtherChar
CropResContents
CropResOtherChar
CropResManagement
CropResManagement_NotSystemShares
CropResManSystem
CropResManSystemCropland
CropResManSystemGrassland
CropResManSystem_WithVolatInApplication
ManureContents
ManureManagement
ManureManSystem
ManureManSystemCropland
ManureManSystemGrassland
ManureManSystem_NoAverageNoTotal
FertApplicCharact

Regions
Regions_2
Countries

FAO_Africa
FAO_Eastern_Africa
FAO_Middle_Africa
FAO_Northern_Africa
FAO_Southern_Africa
FAO_Western_Africa
FAO_Americas
FAO_Northern_America
FAO_Central_America
FAO_Caribbean
FAO_South_America
FAO_Asia
FAO_Central_Asia
FAO_Eastern_Asia
FAO_Southern_Asia
FAO_SouthEastern_Asia
FAO_Western_Asia
FAO_Europe
FAO_Eastern_Europe
FAO_Northern_Europe
FAO_Southern_Europe
FAO_Western_Europe
FAO_Oceania
FAO_Australia_NewZealand
FAO_Melanesia
```

```
FAO_Micronesia
FAO_Polynesia
FAO_EuropeanUnion
SubSaharanAfrica
Match_GeographicAggregates_SingleCountries

FOFA2050_Rest_of_EAP
FOFA2050_Rest_of_HIC
FOFA2050_Rest_of_LAC
FOFA2050_Rest_of_MNA
FOFA2050_Rest_of_SAS
FOFA2050_Rest_of_SSA
FOFA2050_Rest_of_EU
FOFA2050_Rest_of_ECA
Regions_Switzerland
Regions_Austria
Regions_austria_Tal
Regions_Austria_Huegel
Regions_austria_Berg

Activities
Activities_2
Livestock
Livestock_NoAggregates
Ruminants
Poultry
MonogastricsNonPoultry
FishAndSeafood
NonRuminants
Livestock_NoCoupledOutputs
Livestock_CoupledMeatMilk
Livestock_CoupledMeatEggs
Livestock_NonRuminantsWithHerdStructure
Crops
Cereals
Fruits
Treenuts
Pulses
Legumes_NFixing
Crops_NoNFixingLegumes
OilCrops
StarchyRoots
SugarCrops
Vegetables
StimulantsSpices
FibresRubber
Fallows
EnergyCrops
OtherCereals
CitrusFruits
Spices
OtherVegetables
OtherOilcrops
FibresNotCotton
ForageCrops
OtherFodderCrops
GrassActivities
CoreGrassActivities
CoreGrassActivitiesNoTEMPGrass
TempAndPermMeadAndPastures
TempMeadAndPastures
CropsAndCoreGrassActivities
CropsAndTempGrassActivities
SingleCropGrassAndLivestockActivities
FOFA2050_SweetPotato_And_Yams
FOFA2050_Rapeseed_And_Mustardseed
FOFA2050_OtherCrops
FOFA2050_OtherFibreCrops
FOFA2050_OtherFruits
FOFA2050_OtherOilseeds
FOFA2050_OtherRootsAndTubers
FOFA2050_OtherVegetables
FOFA2050_CitrusFruits
FOFA2050_DriedPulses
FOFA2050_OtherCereals
OtherCrops

Commodities
Commodities_2
ForageCommodities
Grasscommodities
ConcentrateCommodities
Commodities_SingleCommodities
Commodities_FeedGroups
Commodities_FeedGroups_DetailedFeedingRations
Commodities_Feed
Commodities_Feed_Grass
Commodities_Feed_NoForageNoGrass
Commodities_Feed_ForageCrops
Commodities_Feed_Cereals
Commodities_Feed_OilCropsAndCakes
Commodities_Feed_Pulses
Commodities_Feed_Roots
Commodities_Feed_Sugar
Commodities_Feed_OthersPlants
Commodities_Feed_OthersAnimals
CommoditiesDAQ_TradeFromToWorld
CommoditiesDAQ_TradeFromToWorldPrimary
CommoditiesDAQ_TradeFromToWorldPrimaryCrops
COmmodities_EquivalentToNonGrassFedAnimalActivityOutput
CommoditiesDAQ_TradeFromToWorldPrimaryAnimals
CommoditiesDAQ_TradeFromToWorldPrimaryAnimals_GrassFed
CommoditiesDAQ_TradeFromToWorldPrimaryAnimals_NonGrassFed
CommoditiesDAQ_TradeFromToWorldDirectlyDerivedFromPrimary
CommoditiesDAQ_TradeFTWorldDirDerivedFromPrimary_Animals
CommoditiesDAQ_TradeFTWDirDerFromPrim_Animals_GrassFed
CommoditiesDAQ_TradeFTWDirDerFromPrim_Anim_NonGrassFed
CommoditiesDAQ_TradeFromToWorldCropCommodAndProducts
CommoditiesDAQ_TradeFromToWorld_FishSeafood
MatchCommAct_CommoditiesDAQ_TradeFromToWorld_CropsPrimary
```

```
MatchCommAct_CommoditiesDAQ_TradeFromToWorld_CropsDerived
MatchPrimDerivedCommodities_DAQ_TradeFromToWorld_CropsDerived
MatchPrimDerivedCommodities_DAQ_TradeFromToWorld_AnimalsDerived
MatchPrimDerivedCommodities_DAQ_TradeFTW_AnimalsDer_GrassFed
MatchCommAct_CommodDAQ_TradeFTWorld_CropCommodAndProducts
MatchCommAct_CommoditiesDAQ_TradeFromToWorld_AnimalsPrimary
MatchCommAct_CommoditiesDAQ_TradeFTW_AnimalsPrimaryGrassFed
MatchCommAct_CommoditiesDAQ_TradeFTW_AnimPrimaryNonGrassFed
MatchCommAct_CommoditiesDAQ_TradeFromToWorld_AnimalsDerived

*the following sets are also needed to read in data
Years
BasisyearsOLD
Basisyears
BasisyearsSeed
Temperatures
Temperatures0to100Celsius
TemperaturesBelow10
TemperaturesAbove28
GreenhouseGases

MineralFertilizerType
MineralFertilizerProdTech
MinFertChar
PopulationGroups
Humans_InputsOutputsOtherCharacteristics
FeedingRationOtherChar
MatchFaostatLiveAnimalItems_Activities
Match_ActivityOutputsToCommodities_Crops
Match_ActivityOutputsToCommodities_Animals
Match_FeedCommoditiesToFeedCommodGroups
Match_FeedCommoditiesToFeedCommodGroups_MainByprodConc
Match_FeedCommodToFeedCommodGroups_DetailedFeedRatio

ConversionLevel
EnergyType
EnergySource
EnergyChar

Scenarios
FOFA2050_Scenarios
BioeSR15_Scenarios

MatchCommAct_AggregateCommodities_Crops
MatchCommAct_AggregateActivities_Crops
MatchCommAct_WellDefinedCoProducts_Crops
MatchCommAct_ComplexCases_Crops
MatchCommAct_ComplexAnimalCommodities
MatchCommAct_CommodEquivalentAct_Crops
MatchCommAct_CommodEquivalentAct_Animals
MatchCommAct_CommodAndProductsEquivalentAct_Crops

SubsetCommod_MatchWithAct_AggregateCommodities
SubsetCommod_MatchWithAct_AggregateActivities
SubsetCommod_MatchWithAct_WellDefinedCoProducts
SubsetCommod_MatchWithAct_ComplexCases
SubsetCommod_MatchWithAct_ComplexAnimalCommodities
SubsetCommod_MatchWithAct_CommodEquivalentAct
SubsetCommod_MatchWithAct_CommodAndProductsEquivalentAct

SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level1_Main
SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level1_Co
SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level2_Main
SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level2_Co
MatchMainWithCoProd_WellDefinedCoProducts_Level1
MatchMainWithCoProd_WellDefinedCoProducts_Level2
MatchMainWithCoPr_WellDefCoProd_Level1And2_TwoByprod
;
```

3.1.72.2 GeneralModelParameters_Inputs.gdx

```
execute_unload 'GeneralModelParameters_Inputs'
*1.1) Activities: input param
ActCropsGrass_Inputs
ActAnimalsAPU_Inputs
ActAnimalsHead_Inputs
ActFishSeafood_Inputs
ActForest_Inputs
ActOthers_Inputs
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-ActAnimalsHead_Inputs
         "Drinking water (m3)"     Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                             CattleTypeInHerd
                                             PigTypeInherd
                                             "AllAndAverageTypes"        "AllProdSyst"        "AllProdCond"      WITH CATTLE AND PIGS!!
         "Total GWP from CED (tCO2e)"
         "Total CED (MJ)"          Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                             CattleTypeInHerd
                                             PigTypeInherd
                                             "AllAndAverageTypes"        "AllProdSyst"
                                                                          "Convent"
                                                                          "Organic"           "AllProdCond"      WITH CATTLE AND PIGS!!
-ActCropsGrass_Inputs
         "N deposition (tN)"       Crops / temp/perm grass                "AllProdSyst"        "AllProdCond"
         "Seeds (t)"               Crops (partly only)                    "AllProdSyst"        "AllProdCond"
         "Total GWP from CED (tCO2e)"
         "Total CED (MJ)           Crops / Core grass activities          "AllProdSyst"
                                       (No MISCANTHUS)                    "Convent"
                                                                          "Organic"           +AllProdCond"
-ActForest_Inputs
         "N deposition (tN)"       "Forest"                                 "AllProdSyst"        "AllProdCond"

$offtext;
```

3.1.72.3 GeneralModelParameters_Outputs.gdx

```
execute_unload 'GeneralModelParameters_Outputs'
*1.2) Activities: output para
ActCropsGrass_Outputs
ActAnimalsAPU_Outputs
ActAnimalsHead_Outputs
ActFishSeafood_Outputs
ActForest_Outputs
ActOthers_Outputs
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-ActAnimalsAPU_Outputs
         "Honey (t)"              "Beehives       "Honey Producing"            "AllProdSyst"        "AllProdCond"
         "Wax (t)"                "Beehives       "Wax Producing"              "AllProdSyst"        "AllProdCond"
-ActAnimalsHead_Outputs
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
         "Milk (t"                                "Milk Producing"             "AllProdSyst"        "AllProdCond"
         "Meat (t)"                               "Meat Producing"             "AllProdSyst"        "AllProdCond"
         "Eggs (t)"                               "Eggs Producing"             "AllProdSyst"        "AllProdCond"
         "HidesSkins (t)"                         "HidesSkins Producing"       "AllProdSyst"        "AllProdCond"
-ActCropsGrass_Outputs
         "MainOutput1 (t)"         Crops/ miscanth / temp/perm grass           "AllProdSyst"        "AllProdCond"
         "MainOutput1 (tDM)"       "Miscanthus" / temp/perm grass              "AllProdSyst"        "AllProdCond"
         "CH4 flooded rice (tCH4)"  "Rice, paddy"                              "AllProdSyst"        "AllProdCond"


$offtext;
```

3.1.72.4 GeneralModelParameters_OtherChar.gdx

```
execute_unload 'GeneralModelParameters_OtherChar'
*1.3) Activities: other chara
ActCropsGrass_OtherChar
ActAnimalsAPU_OtherChar
ActAnimalsHead_OtherChar
ActFishSeafood_OtherChar
ActForest_OtherChar
ActOthers_OtherChar
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-ActAnimalsHead_OtherChar
         "Organic yield gap (ratio org/conv yield"
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                "Milk Producing"           "AllProdSyst         "AllProdCond"
                                                "Meat Producing"           "AllProdSyst         "AllProdCond"
                                                "Eggs Producing"           "AllProdSyst         "AllProdCond"
                                                "HidesSkins Producing"     "AllProdSyst         "AllProdCond"
                                                "Wool Producing"           "AllProdSyst         "AllProdCond"
         "Grass in feed DM (share)"
         "Forage crops in feed DM (share)"
         "Concentrates in feed DM (share)"
         "Residues in feed DM (share)"
         "UE_per_GE (share)"
         "Animal specific FeedGE cont (MJ/t)"
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                "AllAndAverageTypes"       "AllProdSyst"         "AllProdCond"
         "Digestibility of feed (%)"
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                CattleTypeInHerd
                                                PigTypeInherd
                                                "AllAndAverageTypes"       "AllProdSyst"         "AllProdCond"       WITH CATTLE AND PIGS!!
         "Liveweight (t)"
                                 "Cattle"         CattleTypeInHerd
                                 "Pigs"           PigTypeInherd
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens) - NO CATTLE NO PIGS!!
                                                "AllAndAverageTypes"       "AllProdSyst"         "AllProdCond"
         "FeedME_Req_Total (MJ)"
         "FeedXP_Req_Total (t)"
         "FeedGE_Req_Total (MJ)"
                                 "Cattle"         CattleTypeInHerd
                                 "Pigs"           PigTypeInherd
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens) - NO CATTLE NO PIGS!!
                                                "AllAndAverageTypes"       "AllProdSyst"         "AllProdCond"

         "FeedME_Req_MilkProd (MJ)"
         "FeedME_Req_Maintenance (MJ)"
         "FeedME_Req_Walking (MJ)"
         "FeedME_Req_Pregnancy (MJ)"
         "DailyWalkingDistance (km)"
         "Summergrazing (days)"
         "ME to produce 1kg milk (MJ/kg milk)"
         "ME maintenance per weight per day (MJ/kg liveweight/day)"
         "ME maintenance basis per day (MJ/day)"
         "ME walking per km (MJ/km)"
         "ME pregnancy per weight per day (MJ/kg liveweight/day)"
         "ME pregnancy basis per day (MJ/day)"
                                 "Cattle"
                                                "Producing_Dairy_Cattle"  "AllProdSyst"         "AllProdCond"

         "XPperME_InFeedReq (gXP/MJ)"
                                 "Cattle"         CattleTypeInHerd         "AllProdSyst"         "AllProdCond"
                                 "Pigs"           PigTypeInherd            "AllProdSyst"         "AllProdCond"

         "Share organic in total animals (share heads)"
         "Share animal type in total living animals"
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
```

```
                                "Living"                 "AllProdSyst"        "AllProdCond"      WITH CATTLE AND PIGS!!
                                CattleTypeInHerd         "AllProdSyst"        "AllProdCond"
                                PigTypeInherd            "AllProdSyst"        "AllProdCond"
                                "Milk Producing"         "AllProdSyst"        "AllProdCond"
                                "Meat Producing"         "AllProdSyst"        "AllProdCond"
                                "Eggs Producing"         "AllProdSyst"        "AllProdCond"
                                "HidesSkins Producing    "AllProdSyst"        "AllProdCond"
                                "Wool Producing"         "AllProdSyst"        "AllProdCond"


         "N in manure per ton liveweight (tN/t lw/y)"
         "P in manure per ton liveweight (tP2O5/t lw/y)"
                         "Cattle"          CattleTypeInHerd
                         "Pigs"            PigTypeInherd
                         Livestock (NO dairy/beef cattle; layer/broiler chickens) - NO CATTLE NO PIGS!!
                                "AllAndAverageTypes"        "AllProdSyst"        "AllProdCond"
-ActCropsGrass_OtherChar
         "Share organic in total area (share ha)"
         "Organic yield gap (ratio org/conv yield)"
         "N fixation per ton MainOutput1 (tN)"
         "Deforestation (ha)"
         "CultOrgSoils (ha)"
         "CultOrgSoils GHG emissions (tCO2e)"
         "CultOrgSoils C emissions (tCO2e)"
         "CultOrgSoils N2O emissions (tCO2e)"
                         All crops (incl. miscanth.) and grass           "AllProdSyst"        "AllProdCond"
         "Soil water erosion (t soil lost)"
         "Aggreg. Pest. use level (index)"
                         Crops (incl. miscanth.) / temp/perm grass       "AllProdSyst"        "AllProdCond"
$offtext;
```

3.1.72.5 GeneralModelParameters_Various.gdx

```
execute_unload 'GeneralModelParameters_Various'
*1.4) Commodities: nutrient contents
Commod_Contents
Commod_OtherChar

*1.5) Commodity tree paramete
Commod_ProductionShare
Commod_ExtractionRate
Commod_SingleInAggregateCommodShares

*1.6) Crop residues: nutrient
CropResidues_Contents
CropResidues_OtherChar
CropResidues_Management

*1.7) Feeding rations
FeedingRations_Contents
FeedingRations_OtherChar
FeedingRationsHeads_Contents
FeedingRationsHeads_OtherChar
FeedingRationsAPU_Contents

*1.8) Manure: nutrient conten
Manure_Contents
Manure_OtherChar
Manure_Management

*1.9) Fertilizer application:
ManureApplication
CropResAndBiomassApplication
MinFertApplication

*extraction rates:
ExtractionRate_CommodityTree
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-Commod_Contents
         "Calories (kcal)"
         "Protein (t)"
         "N (t)"
         "P2O5 (t)"
         "FeedME (MJ)"
         "FeedGE (MJ)"
         "FeedXP (t)"
         "DM (t)"
         "FeedME in DM (MJ)"
         "FeedGE in DM (MJ)"
         "FeedXP in DM (t)"
                           All/most/many commodities (plant, animal)   "AllProdSyst"        "AllProdCond"
         "N in DM (t)"             "Miscanthus"                        "AllProdSyst"        "AllProdCond"
         "Milk solid contents (t)" "Milk, whole"                       "AllProdSyst"        "AllProdCond"

-Commod_OtherChar
         "Producer price ($)"      Many commodities (plant, animal; NO grass, miscanthus)
                                                                     "AllProdSyst"        "AllProdCond"
         "Util feed (share)"
         "Util food (share)"
         "Util seed (share)"
         "Util processing (share)"
         "Util waste (share)"
         "Util other (share)"
                         All/most/many commodities (plant, animal); incl. grass, miscanthus
                                                                     "AllProdSyst"        "AllProdCond"
-CropResAndBiomassApplication
         "N2O-N per kg fertilizer N applied (tN/tN)"
         "Volatized N as percentage of fertilizer N applied (%)"
         "Leached N as percentage of fertilizer N applied (%)"
         "N2O-N per kg N volatized from fert applic (share)"
         "N2O-N per kg N leached from fert applic (share)"
         "NH3-N as percentage of fertilizer N applied (%)"
                         ALL crop grass activities; incl. ALL grass types; miscanthus
```

```
                                                        "All Residues"         "AllProdSyst"         "AllProdCond"

-CropResidues_Contents
         "N (t) - before management"
         "P2O5 (t) - before management"
         "DM (t) - before management"
                                 Crops / temp/perm grass / miscanth
                                                        "Average residues (t)" "AllProdSyst"         "AllProdCond"

-CropResidues_Management
         "Quantity share in CropResMan system"
                                 Crops / temp/perm grass / miscanth
                                                        "Average residues (t)" CropResManSystems    "AllProdSyst/Con/org        "AllProdCond"
         "Crop res man CH4 (tCH4)"
         "Crop res man N2O (tN2O)"
                                 Crops (NO grass/miscanthus - as their systems come without losses; grass e.g. the N2O is from residues as fertilizers
                                 applied, not from residue management "left on the field")
                                                        "Average residues (t)" CropResManSystems    "AllProdSyst"               "AllProdCond"
         "Crop res man N loss (tN)"
         "Crop res man share P lost (tP2O5/tP2O5 in crop res)"
         "Crop res man P loss (tP2O5)"
                                 Crops / miscanth (NO grass - as their systems come without losses; grass e.g. the N2O is from residues as fertilizers
                                  applied, not from residue management "left on the field")
         "Crop res N for areas (tN)"
         "Crop res P for areas (tP2O5)"
                                 Crops / temp/perm grass / miscanth
                                                        "Average residues (t)" CropResManSystems    "AllProdSyst"               "AllProdCond"

-CropResidues_OtherChar
         "Residue share t DM / t DM MainOutput1 (share)"
                                 Crops / temp/perm grass / miscanth
                                                        "Average residues (t)" "AllProdSyst"         "AllProdCond"

-ExtractionRate_CommodityTree
                                 Crops / temp/perm grass (partly all grass) / miscanth

-FeedingRations_OtherChar
         "Quantity share in DM (share)"
                 "AggregateFeedConcentrates_Commodity"
                 "AggregateFeedForageCrops_Commodity"
                 "AggregateFeedGrass_Commodity"
                 "AggregateFeedResidues_Commodity"
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                 CattleTypeInHerd
                                                 PigTypeInherd
                                                 "AllAndAverageTypes"        "AllProdSyst"        "AllProdCond"      WITH CATTLE AND PIGS!!
-FeedingRationsHeads_OtherChar
         "Percentage GE in feed converted to enteric CH4"
                 "AggregateFeedConcentrates_Commodity"
                 "AggregateFeedForageCrops_Commodity"
                 "AggregateFeedGrass_Commodity"
                 "AggregateFeedResidues_Commodity"
                                 Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                 CattleTypeInHerd
                                                 PigTypeInherd
                                                 "AllAndAverageTypes"        "AllProdSyst"        "AllProdCond"      WITH CATTLE AND PIGS!!

-Manure_Management
         "Bo: max. CH4 prod. cap. (m3CH4/kgVS)"
                                "Cattle"         CattleTypeInHerd
                                "Pigs"           PigTypeInherd
                                Livestock (NO dairy/beef cattle; layer/broiler chickens) - NO CATTLE NO PIGS!!
                                                 "AllAndAverageTypes"        "AllManManSystems"    "AllAndAverageTemp"     "AllProdSyst"
"AllProdCond"
         "Quantity share in ManureMan system"
         "Manure man N volat (% of N in manure)"
         "Manure man N leach (% of N in manure)"
         "Manure man NH3-N (% of N in manure)"
                                "Cattle"         CattleTypeInHerd
                                "Pigs"           PigTypeInherd
                                Livestock (NO dairy/beef cattle; layer/broiler chickens) - NO CATTLE NO PIGS!!
                                                 "AllAndAverageTypes"        ManureManSystem       "AllAndAverageTemp"     "AllProdSyst"
"AllProdCond"
         "MCF: CH4 conversion factor (%)"
         "Manure man N2O dir (tN2O-N/tN)"
         "Manure man N2O-N from N volat (tN/tN volat)"
         "Manure man N2O-N from N leach (tN/tN leach)"
         "Manure man P loss as % of P in manure (%)"
                                Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                 CattleTypeInHerd
                                                 PigTypeInherd
                                                 "AllAndAverageTypes"        ManureManSystem       "AllAndAverageTemp"     "AllProdSyst"
"AllProdCond"      WITH CATTLE AND PIGS!!

-Manure_OtherChar
         "Ash content in feed DM (share)"
                                Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                 CattleTypeInHerd
                                                 PigTypeInherd
                                                 "AllAndAverageTypes"        ManureManSystem       "AllAndAverageTemp"     "AllProdSyst"
"AllProdCond"      WITH CATTLE AND PIGS!!

-MinFertApplication
         "N2O-N per kg fertilizer N applied (tN/tN)"
         "Volatized N as percentage of fertilizer N applied (%)"
         "Leached N as percentage of fertilizer N applied (%)"
         "N2O-N per kg N volatized from fert applic (share)"
         "N2O-N per kg N leached from fert applic (share)"
         "NH3-N as percentage of fertilizer N applied (%)"
                                 ALL crop grass activities; incl. ALL grass types; miscanthus
                                                        "Mineral N fert (N)"        "AllProdSyst"        "AllProdCond"

-ManureApplication
         "N2O-N per kg fertilizer N applied (tN/tN)"
         "Volatized N as percentage of fertilizer N applied (%)"
         "Leached N as percentage of fertilizer N applied (%)"
         "N2O-N per kg N volatized from fert applic (share)"
         "N2O-N per kg N leached from fert applic (share)"
         "NH3-N as percentage of fertilizer N applied (%)"
                                 ALL crop grass activities; incl. ALL grass types; miscanthus
```

```
                                                        Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                                 "AllAndAverageTypes"         "AllProdSyst"        "AllProdCond"

   XXXX ALTERNATIVE XXXX
         ABOVE: besides the first entry, the values are NOT DIFFERENT for different animals - thus may simplify by not having separate but only "All
         Animals" there.
                 Then we have the ALTERNATIVE: it would be as follows:
         "N2O-N per kg fertilizer N applied (tN/tN)"
                                 ALL crop grass activities; incl. ALL grass types; miscanthus
                                                        Livestock (NO dairy/beef cattle; layer/broiler chickens)
                                                                 "AllAndAverageTypes"         "AllProdSyst"        "AllProdCond"
         "Volatized N as percentage of fertilizer N applied (%)"
         "Leached N as percentage of fertilizer N applied (%)"
         "N2O-N per kg N volatized from fert applic (share)"
         "N2O-N per kg N leached from fert applic (share)"
         "NH3-N as percentage of fertilizer N applied (%)"
                                 ALL crop grass activities; incl. ALL grass types; miscanthus
                                                        "All Animals"
                                                                 "AllAndAverageTypes"         "AllProdSyst"        "AllProdCond"

$offtext;
```

3.1.72.6 GeneralModelParameters_Auxiliary.gdx

```
execute_unload 'GeneralModelParameters_Auxiliary'
HumanCharacteristics
SeedContents
MineralFertilizerCharacteristics
GWP_GTP_SOLm
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-HumanCharacteristics
         "kcal/cap/day (ADER) SOFI2012"
         "g protein/cap/day SOFI2012"
         "g fat/cap/day SOFI2012"
         "kcal/cap/day (ADER) Walpole2012"
         "g protein/cap/day Walpole2012"
         "g fat/cap/day Walpole2012"
                 "PopulationAll"           "AllCommodities"

-SeedContents
         "N (t)"
         "P2O5 (t)"
         "DM (t)"
                 Crops (most but not all, e.g. no miscanth.)      "AllProdSyst"        "AllProdCond"

-MineralFertilizerCharacteristics
         "t CO2e/tN production"
                 "mineral N fert (N)"
                 "mineral P fert (P2O5)"
                         "AllMinFertProdTech"        "AllProdSyst"

-GWP_GTP_SOLm               **BIG EXCEPTION!! THIS IS NOT ON COUNTRY LEVEL - just global values
         Values for CO2, CHç, N2O
$offtext;
```

3.1.72.7 GeneralModelVariables_ActivityQuantities.gdx

```
execute_unload 'GeneralModelVariables_ActivityQuantities'
*2.1) Amount of activity units
VActCropsGrass_QuantityActUnits
VActAnimalsAPU_QuantityActUnits
VActAnimalsHead_QuantityActUnits
VActFishSeafood_QuantityActUnits
VActForest_QuantityActUnits
VActOthers_QuantityActUnits
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VActAnimalsAPU_QuantityActUnits
         "Beehives"        "AllAndAverageTypes"    "AllProdSyst"    "AllProdCond"

-VActAnimalsHead_QuantityActUnits
         Livestock (NO dairy/beef cattle; layer/broiler chickens)
                          CattleTypeInHerd
                          PigTypeInherd
                          "Living"                 "AllProdSyst"        "AllProdCond"      WITH CATTLE AND PIGS!!
                          "Milk Producing"         "AllProdSyst"        "AllProdCond"
                          "Meat Producing"         "AllProdSyst"        "AllProdCond"
                          "Eggs Producing"         "AllProdSyst"        "AllProdCond"
                          "HidesSkins Producing    "AllProdSyst"        "AllProdCond"
                          "Wool Producing"         "AllProdSyst"        "AllProdCond"

-VActCropsGrass_QuantityActUnits
         ALL crop grass activities; incl. ALL grass types; miscanthus
                          "AllProdSyst"        "AllProdCond"

-VActForest_QuantityActUnits
         "Forest"
         "Planted forest"
         "Primary forest"   (The previous two categories are SUB-CATEGORIES of the first "Forest" - but NOT EXHAUSTING it)
                          "AllProdSyst"        "AllProdCond"

$offtext;
```

3.1.72.8 GeneralModelVariables_Inputs.gdx

```
execute_unload 'GeneralModelVariables_Inputs'
*2.2) Inputs to activities
VActCropsGrass_Inputs
VActAnimalsAPU_Inputs
VActAnimalsHead_Inputs
VActFishSeafood_Inputs
VActForest_Inputs
VActOthers_Inputs
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VActCropsGrass_Inputs
         "Seeds (t)"              Crops (partly only)                     "AllProdSyst"        "AllProdCond"
$offtext;
```

3.1.72.9 GeneralModelVariables_Outputs.gdx

```
execute_unload 'GeneralModelVariables_Outputs'
*2.3) Outputs from activities
VActCropsGrass_Outputs
VActAnimalsAPU_Outputs
VActAnimalsHead_Outputs
VActFishSeafood_Outputs
VActForest_Outputs
VActOthers_Outputs
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VActAnimalsAPU_Outputs
         "Honey (t)"              "Beehives       "Honey Producing"           "AllProdSyst"        "AllProdCond"
         "Wax (t)"                "Beehives       "Wax Producing"             "AllProdSyst"        "AllProdCond"

-VActAnimalsHead_Outputs
                                Livestock (NO dairy/beef cattle; layer/broiler chickens)
         "Milk (t"                                "Milk Producing"            "AllProdSyst"        "AllProdCond"
         "Meat (t)"                               "Meat Producing"            "AllProdSyst"        "AllProdCond"
         "Eggs (t)"                               "Eggs Producing"            "AllProdSyst"        "AllProdCond"
         "HidesSkins (t)"                         "HidesSkins Producing"      "AllProdSyst"        "AllProdCond"
         "Wool (t)"                               "Wool Producing"            "AllProdSyst"        "AllProdCond"

-VActCropsGrass_Outputs
         "MainOutput1 (t)"      Crops / miscanth / temp/perm grass            "AllProdSyst"        "AllProdCond"
$offtext;
```

3.1.72.10 GeneralModelVariables_OtherChar.gdx

```
execute_unload 'GeneralModelVariables_OtherChar'
*2.4) Other characteristics of a
VActCropsGrass_OtherChar
VActAnimalsAPU_OtherChar
VActAnimalsHead_OtherChar
VActFishSeafood_OtherChar
VActForest_OtherChar
VActOthers_OtherChar
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):

NOT YET ANY DATA IN THIS FILE!!!

$offtext;
```

3.1.72.11 GeneralModelVariables_Various.gdx

```
execute_unload 'GeneralModelVariables_Various'
*2.5) Commodity quantities, nutr
VCommod_Quantity
VCommod_Contents
VCommod_OtherChar

*2.6) Commodity untilizationa
VCommod_Production
VCommod_StockChanges
VCommod_Food
VCommod_Feed
VCommod_Seed
VCommod_Processing
VCommod_Waste
VCommod_Other

VCommod_Food_Contents
VCommod_Feed_Contents
VCommod_Waste_Contents
VCommod_Food_OtherChar
VCommod_Feed_OtherChar
VCommod_Waste_OtherChar

*2.7) Crop residue quantities, n
VCropResidues_Quantity
VCropResidues_Contents
VCropResidues_OtherChar
VCropResidues_Management
```

```
*2.8) Feeding rations quantities
VFeedingRations_Quantity
VFeedingRations_Contents
VFeedingRations_OtherChar

*2.9) Manure quantities, nutrien
VManure_Quantity
VManure_Contents
VManure_OtherChar
VManure_Management

*2.10) Fertilizer application: n
VManureApplication
VCropResAndBiomassApplication
VMinFertApplication
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VCommod_Feed
        Quantities fresh matter used for FEED:
                Crop, livestock and grass commodities; incl. temp/perm grass
                                "AllProdSyst"       "AllProdCond"
-VCommod_Food
        Quantities fresh matter used for FOOD:
                Crop and livestock commodities:
                                "AllProdSyst"       "AllProdCond"
-VCommod_Other
        Quantities fresh matter used for OTHER uses (e.g. bioenergy):
                Crop, livestock commodities; incl. Miscanthus; would also be applicable to temp/perm grass - but currently not; all this is used for
                feed
                                "AllProdSyst"       "AllProdCond"
-VCommod_Processing
        Quantities fresh matter used for PROCESSING:
                Crop, livestock commodities; would also be applicable to miscanthus and temp/perm grass - but currently not; all this is used for
                other uses (energy) and feed
                                "AllProdSyst"       "AllProdCond"
-VCommod_Production
        Quantities fresh matter stemming from DOMESTIC PRODUCTION:
                Crop, livestock commodities; incl. Miscanthus and temp/perm grass
                                "AllProdSyst"       "AllProdCond"
-VCommod_Quantity
        Domestically available QUANTITIES fresh matter (sum of all utilizations, resp. prod + imp - exp):
                Crop, livestock commodities; incl. Miscanthus; temp/perm grass
                                "AllProdSyst"       "AllProdCond"
-VCommod_Seed
        Quantities fresh matter used for SEED:
                Crop and livestock commodities:
                                "AllProdSyst"       "AllProdCond"
-VCommod_StockChanges
        Quantities fresh matter stemming from STOCK CHANGES (can also be negative):
                Crop and livestock commodities:
                                "AllProdSyst"       "AllProdCond"
-VCommod_Waste
        Quantities fresh matter lost as WASTE:
                Crop and livestock commodities:
                                "AllProdSyst"       "AllProdCond"

$offtext;
```

3.1.72.12 GeneralModelVariables_Trade.gdx

```
execute_unload 'GeneralModelVariables_Trade'
*2.11) Import and export quantit
VImportQuantity
VExportQuantity
VImportLivingAnimalsHead
VExportLivingAnimalsHead
*for trade in beehives and other
VImportLivingAnimalsAPU
VExportLivingAnimalsAPU

VImportQuantity_Feed
VExportQuantity_Feed
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VExportLivingAnimalsAPU
        "Beehives"
                exported to "WORLD"
                                        "AllProdSyst"       "AllProdCond"
-VExportLivingAnimalsHead
        All Livestock
                exported to "WORLD"
                        "Living"        "AllProdSyst"       "AllProdCond"
-VImportLivingAnimalsAPU
        "Beehives"
                imported from "WORLD"
                                        "AllProdSyst"       "AllProdCond"
-VImportLivingAnimalsHead
        All Livestock
                imported from "WORLD"
                        "Living"        "AllProdSyst"       "AllProdCond"
-VImportQuantity
        Crop and livestock commodities:
                imported from Countries AND "WORLD"
                                        "AllProdSyst"       "AllProdCond"
-VExportQuantity
        Crop and livestock commodities:
                exported to Countries AND "WORLD"
                                        "AllProdSyst"       "AllProdCond"
-VImportQuantity_Feed
        10 crop commodities
                imported from World to Switzerland
                                        "AllProdSyst"       "AllProdCond"
```

```
-VExportQuantity_Feed
         No data
$offtext;
```

3.1.72.13 GeneralModelVariables_CommodityTree.gdx

```
*2.12) Commodities expressed in primary product equivalents
execute_unload 'GeneralModelVariables_CommodityTree'
VPrimProd_Commod_Quantity
VPrimProd_Commod_Production
VPrimProd_Commod_StockChanges
VPrimProd_Commod_Food
VPrimProd_Commod_Feed
VPrimProd_Commod_Seed
VPrimProd_Commod_Processing
VPrimProd_Commod_Waste
VPrimProd_Commod_Other
VPrimProd_ImportQuantity
VPrimProd_ExportQuantity
VPrimProd_Commod_Quantity_CropActivities
VPrimProd_Commod_Quantity_AnimalActivities
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VPrimProd_Commod_Quantity
         Crop and livestock commodities (prim prod equivalents)
                                         "AllProdSyst"        "AllProdCond"
-VPrimProd_Commod_Quantity_CropActivities
         Crop commodities (prim prod equivalents)
                 Linked to ACTIVITIES   incl AGGREGATE CROP ACTIVITIES  (such as "All Cereals")
                                         "MainOutput1 (t)"
                                                 "AllProdSyst"        "AllProdCond"
-VPrimProd_Commod_Quantity_AnimalActivities
         Livestock commodities (prim prod equivalents)
                 Linked to ACTIVITIES   incl AGGREGATE ANIMAL ACTIVITIES  (such as "All Animals")
                                        "Meat (t)"
                                        "Milk (t)"
                                        "Eggs (t)"
                                        "Honey (t)"
                                        "Wool (t)"
                                                 "AllProdSyst"        "AllProdCond"
$offtext;
```

3.1.72.14 GeneralModelVariables_Auxiliary.gdx

```
execute_unload 'GeneralModelVariables_Auxiliary'
VMineralFertilizerQuantity
VPopulationNumbers
VEnergyProduction
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):
-VMineralFertilizerQuantity
         "Mineral N fert (N)"
         "Mineral P fert (P2O5)"
         "Mineral K fert (K2O)"
                 "AllMinFertProdTech"  "AllProdSyst"
                              also for some WORLD REGIONS
-VPopulationNumbers
         "PopulationAll"
         "Male"
         "Female"

$offtext;
```

3.1.72.15 FOFA2050_BioeSR15_Etc_Data_InModelRunEntities.gdx

```
*store the FOFA2050 and BioeSR15 data
*this is contained in _MR entities, thus store them:
execute_unload 'FOFA2050_BioeSR15_Etc_Data_InModelRunEntities'
ActCropsGrass_Outputs_MR
ActCropsGrass_OtherChar_MR
ActAnimalsHead_Outputs_MR
VActCropsGrass_QuantityActUnits_MR
VActCropsGrass_Outputs_MR
VActCropsGrass_Inputs_MR
VActAnimalsHead_QuantityActUnits_MR
VActForest_QuantityActUnits_MR
VPopulationNumbers_MR
HumanCharacteristics_MR
VEnergyProduction_MR
VCropResidues_Quantity_MR
CropResidues_Management_MR
ActForest_OtherChar_MR
FeedingRationsHeads_OtherChar_MR

AUX_FeedSupplyFactor_BioeSR15_MR
;
$ontext;
from this, you get the following - always on COUNTRY LEVEL (stuff not mentioned is not available!):

*AND - for doing this data, all "Baseline" values that are already available have been assigned - below, THOSE are NOT displayed to be available!!

-ActAnimalsHead_Outputs_MR
         FOFA_BAU_2050
                 Buffaloes, Cattle, Chickens, Goats, Pigs, Sheep
```

```
                        "Milk (t)"      "Milk Producing"            "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"
                        "Meat (t)"      "Meat Producing"            "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"
                        "Eggs (t)"     "Egg Producing"              "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"
-ActCropsGrass_OtherChar_MR
        FOFA_BAU_2050
                Crops (no perm/temp grass/no miscanthus; not all crops)
                        "Cropping intensity (ratio)"                "Irrigated"
                                                                    "Rainfed"        "AllProdCond"
-ActCropsGrass_Outputs_MR
        FOFA_BAU_2050
                Crops (no perm/temp grass/no miscanthus; not all crops)
                        "MainOutput1 (t)"                           "Irrigated"
                                                                    "Rainfed"
                                                                    "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"

        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Miscanthus"
                        "MainOutput1 (t)"
                        "MainOutput1 (t DM)"                        "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"
                "All Cereals"
                "All Sugar Crops"
                "All Oilcrops"
                        "MainOutput1 (t DM)"                        "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"
-ActForest_OtherChar_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Forest"
                        "Forest res bioe N for areas (tN/t DM res)"
                                                                    "AllProdSyst"    "AllProdCond"

-AUX_FeedSupplyFactor_BioeSR15_MR         auxiliary parameter capturing the total feed demand in the BioeSR15 scenario P4
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100

-CropResidues_Management_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Miscanthus"
                        "Average Residues (t)"
                                "Quantity share in CropResMan system"     "Crop res N for areas (tN)"
                        "Average Residues (t DM)"
                                "Crop res N for areas (tN/t DM res)"
                "All crops"
                        "Average Residues (t DM)"
                                "Crop res N for areas (tN/t DM res)"
                                                        "All bioenergy" (crop res man system)
                                                                    "AllProdSyst"
                                                                    "Convent"
                                                                    "Organic"        "AllProdCond"
-HumanCharacteristics_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                 "PopulationAll"
                        "kcal/cap/day (BioeSR15 req)"
                               "All Commodities"
                               "All crop based Commodities"
                               "All animal based Commodities"
                NO COUNTRIES, ONLY Regions:
                                        World
                                        R5ASIA
                                        R5LAM
                                        R5MAF
                                        R5OECD90+EU
                                        R5REF
-VActAnimalsHead_QuantityActUnits_MR
        FOFA_BAU_2050
                "Buffaloes"
                "Cattle"
                "Chickens"
                "Goats"
                "Pigs"
                "Sheep"
                "All animals"  (in LIVESTOCK UNITS - NOT HEADS)
                                 "Living"                  "AllProdSyst"        "AllProdCond"
-VActCropsGrass_Inputs_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Miscanthus"
                "All crops and grass"
                        "N from all fertilizers (tN)"
                                                           "AllProdSyst"        "AllProdCond"
-VActCropsGrass_Outputs_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Miscanthus"
                        "MainOutput1 (t DM)"
                                                           "AllProdSyst"        "AllProdCond"
-VActCropsGrass_QuantityActUnits_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
```

```
        BioeSR15_P4_2100
                 All crop grass activities; incl. perm/temp mead. and past; miscanthus;
                         "AllProdSyst"        "AllProdCond"
        FOFA_BAU_2050
                 All crop activities; NO perm/temp mead. and past; NO miscanthus!!!
                         "Irrigated"
                         "Rainfed"
                         "AllProdSyst"        "AllProdCond"

-VActForest_QuantityActUnits_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Forest"
                "Forestry"
                "Natural forest"   (TO BE CLARIFIED: relation of these categories!!??)
                                 "AllProdSyst"        "AllProdCond"

-VCropResidues_Quantity_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Miscanthus"
                     "Average Residues (t)"
                         "AllProdSyst"        "AllProdCond"
-VEnergyProduction_MR
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "Miscanthus"
                "Forest and crop residues"
                         "Primary Energy"       "AllEnTypes"    "Production (EJ)"
     NO COUNTRIES, ONLY Regions:        World
                                        AFR
                                        CHN
                                        EUR
                                        IND
                                        JPN
                                        LAM
                                        MEA
                                        OAS
                                        ROW
                                        RUS
                                        USA
                         "Secondary Energy"       "AllEnTypes"    "N from Bioe residues (tN)"
                                                                  "Bioe conversion Biomass input (tDM)"  (FOR secondary energy: we have country
values)
                "AllEnSources"
                "Biomass"
                         "Primary Energy"       "AllEnTypes"    "Production (EJ)"
                         "Secondary Energy"     "Electricity"
                                                "Gases"
                                                "Hydrogen"
                                                "Liquids"       "Production (EJ)"
                "EtOH residues"
                         "Secondary Energy"     "Electricity"   "Production (EJ)"
                "Biomass"
                         "Secondary Energy"     "Liquids"       "N from Bioe residues (tN)"
     NO COUNTRIES, ONLY Regions:        World
                                        R5ASIA
                                        R5LAM
                                        R5MAF
                                        R5OECD90+EU
                                        R5REF
-VPopulationNumbers_MR
        FOFA_BAU_2050
        BioeSR15_P4_2010
        BioeSR15_P4_2050
        BioeSR15_P4_2100
                "PopulationAll"

$offtext;
```

3.2 SteeringFile 2

The following describes the content of the code modules executed in “`___V6_SteeringFile2_CoreModelScenariosAndEquations.gms`” in detail. The general structure of this file is described in section 2.2.2. In the following, we shortly list all code modules that are executed and subsequently describe those in detail (the headings displayed are the same as in the structure described in section 2.2.2):

Code modules executed in Steering File 2:

`2) Define sets, parameters and variables and load gdx files from the baseline assignment`

_V6_Sets_GeneralModelSets_ForSteeringFile2
_V6_VariablesAndParameters
_V6_ReadOutputFilesFromSteeringFile1
_V6_VariablesAndParameters_ModelRun_ForSteeringFile2
_V6_BaselineValues_ForModelRuns

3) Run core model equations

_V6_InitialiseSetsForModelRuns

```
$setglobal Scenario "BaselineDerived"
$label Restart
```

_V6_AssignInitialValuesToScenarios
_V6_ScenarioSpecifications
_V6_AssignInitialValuesToScenarios_DifferentiatedProdSystemsEtc
_V6_DataDerived_CropProductionTotalsAndDAQ
_V6_DataDerived_CropResidueManagement
_V6_DataDerived_CropGrassNutrientRequirements
_V6_CoreModelEquations_NutrReqAndFeedSupply_DetailedFeedingRations
_V6_CoreModelEquations_DeriveAnimalNumbAndProd_DetailedFeedingRations
_V6_CoreModelEquations_ManureExcretionAndManagement
_V6_CoreModelEquations_EntericFerm_DetailedFeedingRations
_V6_CoreModelEquations_FertilizerApplication
_V6_CoreModelEquations_FertilizerApplicationEmissions

4) Choice of scenarios

In this part, it is indicated which scenarios shall be run. This has then also to be reflected in the definition of the set `ScenariosResults(Scenarios)`, where only the scenarios of interest are included for the final results output (cf. section 3.2.27).

Example of a scenario choice:

```
$if %Scenario% == "Baseline_NoFCF" $goto EndOfScenarioRuns
$if %Scenario% == "Baseline_100Organic" $setglobal Scenario "Baseline_NoFCF"
$if %Scenario% == "BaselineDerived" $setglobal Scenario "Baseline_100Organic"
```

The following code is always here, governing the scenario loops:

```
$if %RunAllChosenScenarios% == YES $goto Restart
$label EndOfScenarioRuns
```

5) Further calculations after finishing the scenario runs

_V6_DeriveAggregateImpacts_PerUnit
_V6_DeriveTotalImpacts
_V6_DeriveGeographicAggregations
_V6_DeriveActivityGroupAggregations
_V6_DerivePerAPUValues

*The following code is run several times in a loop to get results for the different ways to allocate impacts:

```
$label RestartAllocationCode
```

_V6_DerivePerPrimaryProductImpacts
_V6_DerivePerCommodityImpacts

```
$if %Allocation% == "MainProduct" $goto EndOfAllocationCode
$if %Allocation% == "Price" $setglobal Allocation "MainProduct"
$if %Allocation% == "Protein" $setglobal Allocation "Price"
$if %Allocation% == "Calories" $setglobal Allocation "Protein"
$if %Allocation% == "Mass" $setglobal Allocation "Calories"
$goto RestartAllocationCode
$Label EndOfAllocationCode
```

6) Define some output files

_V6_OutputFiles_SteeringFile2
_V6_ResultsFiles

7) Do some further specific calculations needed for certain aspects
e.g. __SOLmV5_CoreModelEquations_SomeSpecialOutputForNFP69.gms. which is however currently not regularly used as it is rather slow

The following subsections provide detailed descriptions of these code modules.

3.2.1 _V6_Sets_GeneralModelSets_ForSteeringFile2

The first part of this code is identical to the code described in section 3.1.15 describing the code-file "_V6_Sets_GeneralModelSets.gms", defining the same sets, besides the following, which is not needed anymore (Set MatchFaostatLiveAnimalItems_Activities(FAOSTAT_LiveAnimalsItems,Activities)). Then, it defines some further sets, that are also present in Steering File 1, but are defined in other code than "_V6_Sets_GeneralModelSets" that is executed later. These additional sets are the scenario set, and then the many sets related to the commodity trees (for details on those, see section 3.13). In this code-file, all these sets are only defined, but not yet filled with contents – this is then done when loading the data from the output from steering file 1 in section 3.2.3, "_V6_ReadOutputFilesFromSteeringFile1.gms", as all these sets are already present there from the execution of steering file 1.

Scenarios
ATTENTION: do NOT use scenario names with spaces in the name " "; this strangely does not work, e.g. "LfS 2050" causes problems, while "LfS_2050" works without problems. No idea why.

Set containing all scenario names that may be used in the various model runs. The baseline is "Baseline", then there is always a scenario "BaselineDerived" which replicates the baseline and provides additional values needed for the other model runs. Other elements refer to the various scenarios, such as "Baseline_100Organic" which would be the baseline, converted to 100% organic, or "Baseline_NoFCF" which would be the baseline with food-competing feed reduced to zero. The set Scenarios is displayed in the excel-file "SOLm_Documentation_Appendix.xlsx" in the sheet "Scenarios".

FOFA2050_Scenarios(Scenarios)
BioeSR15_Scenarios(Scenarios)

Subsets containing the scenarios used in the FAO 2050 projections (FAO 2018) and in the IPCC SR15 bioenergy scenarios (IPCC 2018).

The following are the sets used for defining the commodity trees, for further details on those, see section 3.13. They divide the set Commodities in 7 subsets (for headings, etc. referred to as Set1-Set7), which have different characteristics regarding the relation of the set elements to single or aggregate commodities, to primary crop products, to co-products, etc. as described below after each set name.

The following sets link the Commodities to Activities and OutputsAnimals or OutputsCropsGrass.

Set1
SubsetCommod_MatchWithAct_AggregateCommodities(Commodities)
MatchCommAct_AggregateCommodities_Crops(Commodities,Activities,OutputsCropsGrass)

Set with the cases where commodities correspond to an aggregate of some non primary commodities. This captures the cases where commodities correspond to an aggregate of some non primary commodities - for prim prod quantities of outputs from activities, we thus need to disaggregate and to translate the commodities to the outputs from activities via commodity trees. Contains crop commodities only, hence only one file needed for crops.

Set2
SubsetCommod_MatchWithAct_AggregateActivities(Commodities)
MatchCommAct_AggregateActivities_Crops(Commodities,Activities,OutputsCropsGrass)

Set with the cases where commodities correspond to prim prod outputs of aggregate activities. This captures the cases where commodities correspond to prim prod outputs of aggregate activities and thus need to be allocated to outputs from single activities. Contains crop commodities only, hence only one file needed for crops.

Set3

SubsetCommod_MatchWithAct_WellDefinedCoProducts(Commodities)

MatchCommAct_WellDefinedCoProducts_Crops(Commodities,Activities,OutputsCropsGrass)

Set matching activities and commodities for the cases of well-defined coproducts. This captures the cases of well-defined coproducts. Contains crop commodities only, hence only one file needed for crops.

Set4

SubsetCommod_MatchWithAct_ComplexCases(Commodities)

MatchCommAct_ComplexCases_Crops(Commodities,Activities,OutputsCropsGrass)

Set matching activities and commodities for the most complex cases such as fats and alcohol". This captures the most complex cases such as fats and alcohol. Contains crop commodities only, hence only one file needed for crops.

Set5

SubsetCommod_MatchWithAct_ComplexAnimalCommodities(Commodities)

MatchCommAct_ComplexAnimalCommodities(Commodities,Activities,OutputsAnimals)

Set matching activities and commodities for the more complex animal commodities. This covers the more complex animal commodities. Contains animal commodities only, hence only one file needed for animals.

Set6

SubsetCommod_MatchWithAct_CommodEquivalentAct(Commodities)

MatchCommAct_CommodEquivalentAct_Crops(Commodities,Activities,OutputsCropsGrass)

Set covering all crop commodities that are equivalent to main outputs from activities

MatchCommAct_CommodEquivalentAct_Animals(Commodities,Activities,OutputsAnimals)

Set covering all animal commodities that are equivalent to main outputs from activities

This captures all commodities that are equivalent to main outputs from activities.

Set7

SubsetCommod_MatchWithAct_CommodAndProductsEquivalentAct(Commodities)

MatchCommAct_CommodAndProductsEquivalentAct_Crops(Commodities,Activities,OutputsCropsGrass)

Set matching activities and commodities for all commodities that are captured together with their derived products and expressed in primary product equivalents.

The following sets capture the product/coproduct relations by defining subsets of all the commodities as follows: main products on level 1, co-products on level 1, main products on level 2, co-products on level 2: via additional sets that capture the main product - co-product pairings (or, in few cases: triplets). This results in the following sets:

SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level1_Main(Commodities)

subset of SubsetCommod_MatchWithAct_WellDefinedCoProducts(Commodities) with the commodities that are on level 1 and that are MAIN Commodities there (e.g. "Starch")

SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level1_Co(Commodities)

subset of SubsetCommod_MatchWithAct_WellDefinedCoProducts(Commodities) with the commodities that are on level 1 and that are CO-Commodities to the main ones there (e.g. "Wheat brans")

SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level2_Main(Commodities)

subset of SubsetCommod_MatchWithAct_WellDefinedCoProducts(Commodities) with the commodities that are on level 2 and that are MAIN Commodities there (e.g. "Gluten")

SubsetCommod_MatchWithAct_WellDefinedCoProducts_Level2_Co(Commodities)

subset of SubsetCommod_MatchWithAct_WellDefinedCoProducts(Commodities) with the commodities that are on level 2 and that are CO-Commodities to the main ones there (e.g. "Wheat brans")

Matching of main and co-products:
MatchMainWithCoProd_WellDefinedCoProducts_Level1(Commodities,Commodities_2)
MatchMainWithCoProd_WellDefinedCoProducts_Level2(Commodities,Commodities_2)
some products have two byproducts, collect them separately:
alias(Commodities,Commodities_3)
MatchMainWithCoPr_WellDefCoProd_Level1And2_TwoByprod(Commodities,Commodities_2,Commodities_3)

3.2.2 _V6_VariablesAndParameters

This is identical to the code described in section 3.1.16.

3.2.3 _V6_ReadOutputFilesFromSteeringFile1

This code file reads the gdx-output-files from Steering file 1. These are the following files (for details on what is contained in each file, see section 3.1.72):

- GeneralModelSets.gdx
- GeneralModelParameters_Inputs.gdx
- GeneralModelParameters_Outputs.gdx
- GeneralModelParameters_OtherChar.gdx
- GeneralModelParameters_Various.gdx
- GeneralModelParameters_Auxiliary.gdx
- GeneralModelVariables_ActivityQuantities.gdx
- GeneralModelVariables_Inputs.gdx
- GeneralModelVariables_Outputs.gdx
- GeneralModelVariables_OtherChar.gdx
- GeneralModelVariables_Various.gdx
- GeneralModelVariables_Trade.gdx
- GeneralModelVariables_CommodityTree.gdx
- GeneralModelVariables_Auxiliary.gdx

The following file is read later only, after having introduced the "..._MR" parameters and variables, i.e. in module "_V6_VariablesAndParameters_ModelRun_ForSteeringFile2" (see section 3.2.4), section 5.0.

- FOFA2050_BioeSR15_Etc_Data_InModelRunEntities.gdx

3.2.4 _V6_VariablesAndParameters_ModelRun_ForSteeringFile2

This file declares the parameters and variables for the model runs. In this, it is largely identical to the file "`_V6_VariablesAndParameters_ModelRun.gms`" described in section 3.1.68 (but the set scenario is not defined here but loaded from the output of SteeringFile1). It then also loads the yet missing data from "FOFA2050_BioeSR15_Etc_Data_InModelRunEntities.gdx" (section 5.0 in the code file).

After having defined and loaded the model run parameters and variables, those for which baseline values are available are assigned so, i.e. so that the model run parameter and variables with scenario dimension "Baseline" are set equal to the baseline values as read in the previous files (sections 5.1) ff in the code file).

At the end of this file, a number of sets and some few parameters for the footprint calculations are defined. Originally, they have been specified in _V6_DerivePerPrimaryProductImpacts and _V6_DerivePerCommodityImpacts only, but later, the option was added to switch the footprint calculations on or off to safe calculation time – but some of these sets are also used elsewhere, hence the relocation of their definition into this file.

However, before this initialization, it also adds a few additional parameters and variables as follows, mainly used to fasten code execution in certain places:

In the section *1.7) Feeding rations it has the following in addition:

FeedingRationsHeads_Contents_MR(Regions,Activities,AnimalTypeInHerd,Commodities,Contents,ProductionSystems,ProductionConditions,Scenarios)

"nutrient contents of feed/Feeding rations - UNIT: units nutrient per animal head"

In the section *2.6) Commodity utilizations, it adds

AUX_VCommod_Feed_Contents_MR(Regions,Commodities,Contents,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable - used to fasten some divisions - total nutrient contents of commodity used for feed - UNIT: total units nutrient

In the section *3) Auxiliary parameters, it adds

AUX_FeedSupplyFactor_BioeSR15_MR(Regions,Scenarios)

auxiliary parameter capturing the total feed demand in the BioeSR15 scenario P4

AUX_ActAnimalsHead_OtherChar_MR(Regions,Activities,AnimalTypeInHerd,OtherCharAnimals,ProductionSystems,ProductionConditions,Scenarios)

auxiliary parameter for some calculations related to other characteristics of the animal activities"

In the section *4) Auxiliary variables, it adds

AUX_Inverse_VActCropsGrass_QuantityActUnits_MR(Regions,Activities,ProductionSystems,ProductionConditions,Scenarios)

an auxiliary variable - the INVERSE of the total amount of activity units - UNIT: 1 divided by Number of hectares

AUX_Inverse_VActAnimalsHead_Outputs_MR(Regions,Activities,AnimalTypeInHerd,OutputsAnimals,ProductionSystems,ProductionConditions,Scenarios)

an auxiliary variable - the INVERSE of the total outputs from the animal activities undertaken - UNIT: 1 divided by total output

AUX_VCommod_Production_MR(Regions,Commodities,ProductionSystems,ProductionConditions,Scenarios)

"An auxiliary version of total quantity of commodity production - used in the DAQ etc. calculations - UNIT: tons"

AUX_Inverse_VCommod_Quantity_MR(Regions,Commodities,ProductionSystems,ProductionConditions,Scenarios)

an auxiliary variable - the INVERSE of DAQ - UNIT: 1 divided by quantity

AUX_Inverse_VExportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

another auxiliary variable for the DAQ calculations

AUX_ProdPlusImport_VCommod_Production_MR(Regions,Commodities,ProductionSystems,ProductionConditions,Scenarios)

a further auxiliary variable used for the DAQ calculations - sum of imports and production of certain commodities

AUX_Inverse_VFeedingRations_Quantity_MR(Regions,Activities,AnimalTypeInHerd,Commodities,ProductionSystems,ProductionConditions,Scenarios)

an auxiliary variable - the INVERSE of the total quantity of feed - UNIT: tons

VImportStorageAUX_VImportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable for fastening the code: total crop commodity quantity IMPORTED into Regions FROM Regions_2 - UNIT: tons

VExportStorageAUX_VExportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable for fastening the code: total crop commodity quantity EXPORTED from Regions INTO Regions_2 - UNIT: tons

And for animal commodities:

VImportStorageAUX2_VImportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable for fastening the code: total animal commodity quantity IMPORTED into Regions FROM Regions_2 - UNIT: tons

VExportStorageAUX2_VExportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable for fastening the code: total animal commodity quantity EXPORTED from Regions INTO Regions_2 - UNIT: tons

And some further auxiliary variables are needed to avoid corrupting some calculations when calculating some intermediate values of interest:

AUX_VImportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable allowing to calculate some intermediate values of interest without corrupting other code

AUX_VExportQuantity_MR(Regions,Regions_2,Commodities,ProductionSystems,ProductionConditions,Scenarios)

AUXILIARY variable allowing to calculate some intermediate values of interest without corrupting other code

3.2.5 _V6_InitialiseSetsForModelRuns

This file contains the initialisation of the core model sets for the model runs. It thus defines which regions to use, which set of activities and commodities, etc. This is governed by using subsets in the dimensions of the parameters and variables only. E.g. when calculating on country level only, the code does not need the full set Regions, but the subset Countries is enough. Then an assignment of Regions_MR to be equal to Countries is made. Or similarly when using a NUTS2-level resolution for the EU, a subset of Regions containing all countries outside the EU and all NUTS2-regions in the EU is used. Thus, as with the parameters and variables, the sets for the model runs are defined with an suffix "_MR" and all model equations use those sets, and these sets are then allocated with the `alias`-statement, for example, or defined directly. Currently, this reads as follows:

```
alias(Activities_MR,SingleCropGrassAndLivestockActivities);
alias(Crops_MR,Crops);
alias(CoreGrassActivities_MR,CoreGrassActivities);
alias(CropsAndCoreGrassActivities_MR,CropsAndCoreGrassActivities);
alias(CropsAndTempGrassActivities_MR,CropsAndTempGrassActivities);
alias(CoreGrassActivitiesNoTEMPGrass_MR,CoreGrassActivitiesNoTEMPGrass);
alias(Regions_MR,Countries);
alias(Regions_MR_2,Regions_MR);
*alias(Commodities_MR,Commodities_SingleCommodities);
        *some commoditiees for the trade calculations are aggregates - hence use all commodities
alias(Commodities_MR,Commodities);
alias(Commodities_MR_2,Commodities_MR);
alias(CommoditiesDAQ_TradeFromToWorld_MR,CommoditiesDAQ_TradeFromToWorld);
alias(ConcentrateCommodities_MR,ConcentrateCommodities);
alias(ForageCommodities_MR,ForageCommodities);
alias(GrassCommodities_MR,GrassCommodities);

alias(Livestock_MR,Livestock);
alias(AnimalTypeInHerd_MR,AnimalTypeInHerd);
```

This file also contains some auxiliary sets needed to assign baseline values to scenario values in the code file described in section 3.2.6 below.

Currently, these are the following:

*the following is used for adequately assigning the values from the parameter ManureManagement to the scenarios (see code-file AssignInitialValuesToScenarios, section 1.10):

Set ManureManagement_NotContainingBo(ManureManagement)

Set ManureManagement_ForDerivingTotals(ManureManagement)

"Contains the parameters related to manure management that are needed and possible to use for deriving totals - e.g. emissions, N losses, etc.; UNITS - per t manure TS DM"

Set OtherCharCropsGrass_ForAggregation(OtherCharCropsGrass)

"Characteristics of the crop or grass activity - USED FOR AGGREGATION - being undertaken that cannot be captured well by mass/nutrient flows; this can be the biodiversity loss or deforestation, but also monetary flows, etc."

3.2.6 _V6_BaselineValues_ForModelRuns

This file contains the derivations of certain baseline values needed for the model runs, as described below (structure copied from the code file):

`- 1) Do some Baseline calculations that are needed for the scenarios (e.g. to derive single crop shares in total crop groups, etc.)`

`1.1) Total crop production`

`1.2) Commodities`

Production of primary products as commodities

Use the sets Match_ActivityOutputsToCommodities_Crops(Activities,OutputsCropsGrass,Commodities) to link production from activities to primary commodities PRODUCTION level (not DAQ) - equal the activity outputs to these commodities by these matching sets. Derive DAQ by adding imports, subtracting exports and accounting for stock changes. These are then the quantities for the primary products, - all processed (e.g. wheat flour from wheat) can then be derived from this basis.

`1.3) derive some sums of activities that we need in the baseline`

these sums are needed for deriving imp/exp values in the scenarios - they are used for scaling, not for aggregate quantities - hence the partly maybe not fully correct assignments (summing over all fruits for fruits fresh, nes, etc.

`- 2) Nutrient requirements`

Derive the feed requirements of dairy cows for the baseline by means of the same equations as done later for the scenarios (using energy for maintenance, walking, pregnancy, etc.); derive some other relevant feed requirement quantities.

`- 3) Utilization of DAQ and Some further commodity-related calculations`

Derive the utilization quantities for the baseline from DAQ and utilization shares

`- 4) Feed supply`

`4.1) Feed supply from DAQ`

Derive total quantities per utilization; on the level of domestically available quantities, and based on DAQ and utilization shares; another path of calculations is to start from these quantities and then to derive the shares rather via division by total DAQ. All this is in fresh matter. Thus, here: derive Feed from DAQ*FeedUtilization, and DAQ is Prod+Imp-Exp as also derived in the TotalProdDAQ-module (i.e. in "_V6_DataDerived_CropProductionTotalsAndDAQ.gms" and similarly for animals). Then derive some further feed related quantities, such as the sum over commodities to get the total supply per feed group.

`4.2) Feed supply demand ratio`

Derive the ration of feed supplied and demanded per country and feed group for the baseline.

`4.3) Share of animals in requirements per feed group`

Derive the share of animals in total feed requirements per feed group, for the baseline.

`4.4) Calculations for detailed feeding rations calculations`

Various calculations needed for working with detailed feeding rations (for details, see the code)

`- 5) Suckled animals`

Derive the ratio of suckling to suckled animals to derive the number of the latter from the former, as the latter do not report any feed requirements (being included in the requirements from the suckling animal), for the baseline

`- 6) Values needed to adjust mineral fertilizer quantities in the scenarios`

Derive some baseline values that are needed to adjust fertilizer quantities in the scenarios (see section 3.2.17 for how this is done in detail).

`- 7) Some simplification of fertilizer application emission values`

the parameter ManureApplication_MR gets too big - as it is not differentiated per crops, replace the Activities by "All crops", first assigning it from one crop and choose another in case the first is not grown in a region

`- 8) Corrections of some errors`

Here, errors in the data can be corrected (e.g. wrong units by a factor of 1000 in an entry from FAOSTAT), for details see the code

3.2.7 _V6_AssignInitialValuesToScenarios

This file contains the code for assigning baseline values to all parameters for initialising the scenario runs. This is then later replaced by more specific values, if needed, - as provided in the scenario specification file. For all parameters, where values are missing for the scenario, these are assigned here equal to the baseline. Variables are all derived during the code execution, or provided by the scenario definitions, hence there is no initialisation of variables with baseline values needed.

3.2.8 _V6_ScenarioSpecifications

ATTENTION: do NOT use scenario names with spaces in the name " "; this strangely does not work, e.g. "LfS 2050" causes problems, while "LfS_2050" works without problems. No idea why.

This file contains the set of assumptions for each of the scenarios, each one labelled with a label related to its scenario name. The execution of the respective part for the scenario chosen for a specific scenario run is governed by the global variable `%Scenario%` (cf. section 3.2 at the beginning, where it reads "4) Choice of Scenarios…"). This is organized at the beginning by the following code:

```
$if %Scenario% == "Baseline" $goto AssumptionsBaseline
$if %Scenario% == "BaselineDerived" $goto AssumptionsBaselineDerived
$if %Scenario% == "BioeSR15_P4_2050" $goto AssumptionsBioeSR15_P4_2050
$if %Scenario% == "BioeSR15_P4_2050_RefNoBioe" $goto AssumptionsBioeSR15_P4_2050_ReferenceNoBioe
$if %Scenario% == "BioeSR15_P4_2100" $goto AssumptionsBioeSR15_P4_2100
$if %Scenario% == "BioeSR15_P3_2050" $goto AssumptionsBioeSR15_P3_2050
$if %Scenario% == "BioeSR15_P4_2050_Bio" $goto AssumptionsBioeSR15_P4_2050_Bio
$if %Scenario% == "BioeSR15_P4_2050_Bio_AreaIncrease" $goto AssumptionsBioeSR15_P4_2050_Bio_AreaIncrease
$if %Scenario% == "BioeSR15_P4_2050_Bio_AreaIncr_NoFCF" $goto AssumptionsBioeSR15_P4_2050_Bio_AreaIncr_NoFCF
$if %Scenario% == "BioeSR15_P4_2050_Bio_AreaIncr_NoFCF_LessFW" $goto AssumptionsBioeSR15_P4_2050_Bio_AreaIncr_NoFCF_LessFW

$if %Scenario% == "Baseline_100Organic" $goto AssumptionsBaseline_100Organic
$if %Scenario% == "Baseline_NoFCF" $goto AssumptionsBaseline_NoFCF
```

etc.…

Some further details and examples on how to specify scenarios are given in section 3.4.

In general, the following parameters and variables from various code files **(most likely) need to be specified** in the scenario specifications, while all other parameters and variables can (if required by the scenario) but need not be specified specifically for the scenarios.

From section 3.2.10:
To run any calculation with SOLm,
1) the areas harvested of the various crops and grasslands, etc.:

```
VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Activities_MR,
        ProductionSystems,ProductionConditions,"%Scenario%")
```

need to be specified in the scenario specifications as **these variables ARE NOT AVAILABLE as defaults**. All parameters are available as defaults via the initialization (cf. section 3.2.6, but also they can be changed in the scenario specifications, if needed) and all other variables are then derived.

From section 3.2.13:
Many scenarios will address changes in feeding rations, hence the following parameters for which there are default values available often will **need to be re-specified at least partly – according to the scenario** in the scenario specifications (all other values are available via the initialization (cf. section 3.2.6); but also they can be changed in the scenario specifications, if needed).

2) DM quantity shares for each commodity in per head commodity-specific feeding rations

```
FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,
      AnimalTypeInHerd,Commodities_Feed,
            "Quantity share in DM (share)",
                  ProductionSystems,ProductionConditions,"%Scenario%")
```

And, maybe, the total per head GE requirements need to be adjusted in the course of this as well, i.e. the following needs re-specification (but also here, as said, default values are available).

3) Total GE feed requirement per head from ALL feed

```
ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
            "FeedGE_Req_Total (MJ)",
                  ProductionSystems,ProductionConditions,"%Scenario%")
```

From section 3.2.14

Furthermore, when changing the feeding rations, we need to assure that the share in total feed commodities available by the animals is adjusted accordingly, if needed. Thus, the following may need to be specified in the scenario specifications as well (or also for other types of scenarios, where e.g. a reduction of feed supply without changing scenarios is analyzed, etc.)

4) share of the specific animal type in total quantity feed requirements of all animals, for all grass commodities aggregated and for all cereal commodities aggregated (as those two are central for the derivation of animal numbers in the current version of SOL; Cereals play a special role as key concentrate feed that is then used to derive the non-grass-fed animals). The element "Quantity share in total feed req of all animals" is currently used and available only for the aggregate grass and cereal commodities, not for single commodities, hence it has to be directly specified at this level of aggregation in the scenario specifications, if others than the baseline values should be used.

Importantly, changes to this parameter are however **NOT NECESSARY** when changing the feeding ratios. **It is well consistent to use the relevant baseline values**. If, e.g. the feeding ration fro grass for cattle is increased, this does not mean that cattle needs to become a larger share of total available grass quantities. If this latter share is not changed, and total grass supply is not changed either, it means that cattle gets the same quantity as in the baseline, thus allowing to grow a LOWER number of cattle. Other ruminants also get the same grass feed supply as before and thus result in the same numbers as in the baseline. Similar for cereals: after having derived ruminant numbers based on grass supply, the remaining quantity of cereals is determined – which is not affected by the share of cereals allocated to a specific animal species This remaining cereal quantity is then allocated to non-grass-fed animals according to the relative remaining shares in total cereal requirements.

Hence the factor

FeedingRations_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"AggregateFeedCereals_Commodity","Quantity share in total feed req of all animals",ProductionSystems,ProductionConditions,"%Scenario%") / FeedingRations_OtherChar_MR(Regions_MR,"AnimalsWithoutGrassInFeedingRation","AllAndAverageTypes","AggregateFeedCereals_Commodity","Quantity share in total feed req of all animals",ProductionSystems,ProductionConditions,"%Scenario%")

when deriving non-grass-fed animal numbers. Changes in these shares are thus to be implemented specifically in the scenario assumptions, not in consequence of formal requirements due to changes in feeding rations, but in consequence of a change in assumptions on how much feed one species should get in relation to another, i.e. in scenarios, where there is an interest in addressing consequences from a shift from pig to chicken meat production.

Importantly, if changes in these parameters are done, those changes have to be consistent over all animals and both grass and cereal aggregates – i.e. the sum of shares over all animals has to equal 1.

```
FeedingRations_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
      "AggregateFeedGrass_Commodity",
            "Quantity share in total feed req of all animals",
                  ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRations_OtherChar_MR(Regions_MR,Livestock_MR,AnimalTypeInHerd,
      "AggregateFeedCereals_Commodity",
```

```
            "Quantity share in total feed req of all animals",
ProductionSystems,ProductionConditions,"%Scenario%")
```

Furthermore, the following variables are governed by default values/calculations that may be changed depending on the scenario chosen. The second of these is of minor importance for the common food systems scenarios only, unless it explicitly gets linked to yields or such (pollination services), which is currently not the case.

5) DAQ of fish and seafood commodities, as these are not yet internally determined and are just set equal to the baseline value per default, if not replaced by other values

```
VCommod_Quantity_MR.l(Regions_MR,
      CommoditiesDAQ_TradeFromToWorld_FishSeafood,
            ProductionSystems,ProductionConditions,"%Scenario%")
```

6) Number of beehives, as these are internally determined by some gross proportionality to pollination services demand, depending on the cultures that require such – but this default may often be decided to be replaced by other values via scenario specifications

```
VActAnimalsAPU_QuantityActUnits_MR.l(Regions,Activities,AnimalTypeInHerd,
      ProductionSystems,ProductionConditions,"%Scenario%")
```

From section 3.2.17:

Finally, it is likely that mineral fertilizer quantities are changed in scenarios and other than default values should be used. If so, the following variables need to be specified

7) the mineral N and P fertilizer quantities

```
VMineralFertilizerQuantity_MR.l(Regions_MR,
      "mineral N fert (N)","AllMinFertProdTech","AllProdSyst",
            "%Scenario%")

VMineralFertilizerQuantity_MR.l(Regions_MR,
      "mineral P fert (P2O5)","AllMinFertProdTech","AllProdSyst",
            "%Scenario%")
```

and also, if differentiation between org/convent is done, the following then needs ot be adjusted:

```
VMineralFertilizerQuantity_MR.l(Regions_MR,
      "mineral N fert (N)","AllMinFertProdTech","Convent",
"%Scenario%")
```

3.2.9 _V6_AssignInitialValuesToScenarios_DifferentiatedProdSystemsEtc

For the scenarios, it is possible to choose whether organic and conventional (or for other systems, such as rainfed/irrigated, etc.) values shall be used or not. If so, these values are initialized in this module. This choice is governed by setting it accordingly in the scenario specifications, i.e. by choosing a global variable there such as to then execute this specific code or not, cf. section 3.4).

This differentiation is executed conditional to such values already being available, e.g. from the scenario specification or from specific data read in SteeringFile1.

In the scenario specifications, some values may be assumed to be zero - this is however only possible by having set them equal to "eps", as otherwise, they would have been treated as being missing by GAMS and thus replaced by baseline values in the code above and the previous module. Hence, at the end of this module, all eps-values are thus set to zero again, after these assignments.

3.2.10 _V6_DataDerived_CropProductionTotalsAndDAQ

This file contains the calculations to derive total crop production (areas*yields) and DAQs (Production + Imports - Exports) and their utilization as well as some further relevant variables and parameters. Animal products are derived later, after having calculated animal numbers. The table of contents is:

```
$ontext;
- 1) Total crop production
- 2) Commodities
         2.1) Production of primary products as commodities
         2.2) Productiion, Import, export and DAQ values for commodities that are not primary
               products as commodities
                 2.2.1) Commodities that directly correspond to primary products from
                      activities but are mostly named differently
                 2.2.2) Derived commodities that are directly derived from primary commodities
                      for which trade and production data and thus DAQ are available
                 2.2.3) Crop commodities that are pooled from primary commodities plus derived
                      products
                 2.2.4) Remaining complex commodities (beer, sweeteners, beverages)
- 3) Adjust total exports such as to be in the same relation as in the baseline:
- 4) Utilization of DAQ and Some further commodity-related calculations
$offtext;
```

In more detail:

```
- 1) Total crop production
```

It derives `"Land use per Mainoutput1 (ha)"` as the inverse of yields as a new parameter per ton output:

```
ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"Land use per Mainoutput1
        (ha)",ProductionSystems,ProductionConditions,"%Scenario%")
                  $ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"MainOutput1
                       (t)",ProductionSystems,ProductionConditions,"%Scenario%")
          = 1/ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"MainOutput1
               (t)",ProductionSystems,ProductionConditions,"%Scenario%");
```

Then it derives the total output by multiplying the total area with yields:

```
VActCropsGrass_Outputs_MR.l(Regions_MR,Activities_MR,OutputsCropsGrass,ProductionSystems,Produ
               ctionConditions,"%Scenario%")
        = VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Activities_MR,ProductionSystems,
               ProductionConditions,"%Scenario%")
        *ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,OutputsCropsGrass,ProductionSystems
               ,ProductionConditions,"%Scenario%");
```

And it derives total input values, as far as available – e.g. seed inputs, by multiplying total area with inputs per hectare:

```
VActCropsGrass_Inputs_MR.l(Regions_MR,Activities_MR,InputsCropsGrass,ProductionSystems,
               ProductionConditions,"%Scenario%")
= VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Activities_MR,ProductionSystems,
               ProductionConditions,"%Scenario%")
        *ActCropsGrass_Inputs_MR(Regions_MR,Activities_MR,InputsCropsGrass,
               ProductionSystems,ProductionConditions,"%Scenario%");
```

```
- 2) Commodities
        2.1) Production of primary products as commodities
```

Commodity production of primary products is set equal to the domestic production from the activities, by means of the matching file between commodities and activities:

```
VCommod_Production_MR.l(Regions_MR,Commodities_MR,ProductionSystems,ProductionConditions,
               "%Scenario%")
        = sum((Activities_MR,OutputsCropsGrass)
               $Match_ActivityOutputsToCommodities_Crops(Activities_MR,OutputsCropsGrass,
                      Commodities_MR),
               VActCropsGrass_Outputs_MR.l(Regions_MR,Activities_MR,OutputsCropsGrass,
                      ProductionSystems,ProductionConditions,"%Scenario%"));
```

```
        2.2) Production, Import, export and DAQ values for commodities that are not primary
                products as commodities
$ontext;
*here, the prod, imp, exp, DAQ values are calculated for the commodities as listed in Set
        CommoditiesDAQ_TradeFromToWorld(Commodities)
*these commodities come in three types, captured in three sets, and some corresponding
        matching sets needed in the code:
        Set CommoditiesDAQ_TradeFromToWorldPrimary(Commodities) "Commodity set - only PRIMARY
commodities that are produced in a country from an activity only, not from imported
commodities - used for the basic trade calculations ...cf. explanations to the other  sets
above"
        Set CommoditiesDAQ_TradeFromToWorldDirectlyDerivedFromPrimary(Commodities) Derived
commodities that are directly derived from primary commodities for which trade and production
data and thus DAQ are available
        Set CommoditiesDAQ_TradeFromToWorldCropCommodAndProducts(Commodities) "crop
commodities that are pooled from primary commodities plus derived products"
Corresponding matching sets:
        Set
MatchCommAct_CommoditiesDAQ_TradeFromToWorld_CropsPrimary(Commodities,Activities,OutputsCropsG
rass) "Matching set to match the crop commodities used for the basic trade calculations with
the corresponding activities - used to derive the change in production of these commodities,
if the production of the activities changes, for example"
        Set
MatchCommAct_CommoditiesDAQ_TradeFromToWorld_CropsDerived(Commodities,Activities,OutputsCropsG
rass) "Matching set to match the crop commodities used for the basic trade calculations with
the corresponding activities - used to derive the change in production of these commodities,
if the production of the activities changes, for example"
        Set
MatchPrimDerivedCommodities_DAQ_TradeFromToWorld_CropsDerived(Commodities,Commodities_2)
"Matching the derived commodities that are derived from primary commodities as captured in the
previous set - DAQ and trade calculations for them are different than for the other derived
commodities)
        Set
MatchCommAct_CommodDAQ_TradeFTWorld_CropCommodAndProducts(Commodities,Activities,OutputsCropsG
rass) "Matching set to match the crop commodities that are pooled from primary commodities
plus derived products with the underlying activity"
        $offtext;

*the import/export values are done on the basis of to/from WORLD values - as we are interested
in the DAQs in the main model code, not in the origin of the commodities. The origin is used
later after the main model run when deriving impacts embodied in imported commodities

*animal commodities are derived later, when the animal production is available
```

Sections 2.2.1 to 2.2.4 in the code then describe how production, imports, exports and DAQ for these commodities are derived in the scenarios. The basic idea is to scale these by the changes in production of the underlying primary product from the activities, or of aggregates of such, if it is not directly assignable (as e.g. beer that can be derived from a number of activities). Commodity imports scale with total world production, commodity exports with domestic production, and commodity production with domestic production of underlying commodities plus imports. Depending on the type of commodities, this is handled differently, as detailed in the code.

3) Adjust total exports such as to be in the same relation as in the baseline:

This code is used to adjust total global exports such as to have the same ratio over total global imports as in the baseline. This is needed in case imports are specified in the scenario specifications, as this does not fully translate to accordingly changed exports. DAQ of exporting countries is then accordingly adapted as well.

4) Utilization of DAQ and Some further commodity-related calculations

Utilization quantities are then derived from DAQ times utilization shares (either taken from the baseline or specified in the scenario specification) and nutrient and other quantities of DAQ are derived by multiplying DAQ with the respective commodity nutrient contents. See the code file for further details.

To run these calculations, the following inputs **need to be specified** in the scenario specifications (all other values are available via the initialization (cf. section 3.2.6); but also they can be changed in the scenario specifications, if needed).

- the areas harvested of the various crops and grasslands, etc.:
VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Activities_MR, ProductionSystems,ProductionConditions,"%Scenario%")

3.2.11 _V6_DataDerived_CropResidueManagement

This file contains the code to model crop residue management and related emissions. The table of contents is:

```
$ontext;
- 1) Crop residue management
         1.1) Derive nutrients available for areas
         1.2) Derive residue quantities:
         1.3) Derive other nutrient such as N and P2O5 contents of residues:
         1.4) Crop res management characteristics from total residues
$offtext;
```

In more detail:

```
- 1) Crop residue management
       1.1) Derive nutrients available for fertilization of areas
```

this is the reminder of N in crop residues, after subtracting losses:
N losses corresponding to N2O emissions:

```
CropResidues_Management_MR(Regions_MR,Activities_MR,"Average Residues (t)",
        "Crop res man N loss (tN)",CropResManSystem,ProductionSystems,
                ProductionConditions,"%Scenario%")
          = CropResidues_Management_MR(Regions_MR,Activities_MR,"Average Residues (t)",
                "Crop res man N2O (tN2O)",CropResManSystem,ProductionSystems,
                ProductionConditions,"%Scenario%")*28/44;
*for "Open burning", all N is lost:
CropResidues_Management_MR(Regions_MR,Activities_MR,"Average Residues (t)",
        "Crop res man N loss (tN)","Open burning",ProductionSystems,
                ProductionConditions,"%Scenario%")
          = CropResidues_Contents_MR(Regions_MR,Activities_MR,"Average residues (t)",
                "N (t) - before management",ProductionSystems,ProductionConditions,
                        "%Scenario%");
```

Then derive the N that is available to be put on the fields by subtracting these losses from the nitrogen available.
For residue management "feed", no N is lost, but nothing remains for fertilization.
To avoid mistakes, set negative values equal to zero.
Similarly, derive the P available for fertilization via residues.
See the code for details.

```
         1.2) Derive residue quantities:
```

This code calculates the residue production. This is derived based on the dry matter (DM) production, using the residue shares of tons DM residues per ton DM production - the latter is available on commodity level, namely as “`VCommod_Contents_MR.l(Regions,Commodities,"DM (t)", ProductionSystems,ProductionConditions,"%Scenario%")`”. Thus, derive residue quantities again with the general activity-commodity-matching:

```
VCropResidues_Contents_MR.l(Regions_MR,Activities_MR,"Average residues (t)",
        "DM (t) - before management",ProductionSystems,ProductionConditions,"%Scenario%")
  = sum(Commodities_MR$Match_ActivityOutputsToCommodities_Crops(Activities_MR,
```

```
        "MainOutput1 (t)",Commodities_MR),
      VCommod_Contents_MR.l(Regions_MR,Commodities_MR,"DM (t)",ProductionSystems,
             ProductionConditions,"%Scenario%")
      *CropResidues_OtherChar_MR(Regions_MR,Activities_MR,"Average residues (t)",
             "Residue share t DM / t DM MainOutput1",ProductionSystems,ProductionConditions,
             "%Scenario%"));
```

Then derive the wet matter residue quantities by division with DM contents (the "before management" values is used to indicate that these quantities may reduce during management due to various losses and emissions and that the quantities applied to the fields are thus lower):

```
VCropResidues_Quantity_MR.l(Regions_MR,Activities_MR,"Average residues (t)",ProductionSystems,
      ProductionConditions,"%Scenario%")
       $CropResidues_Contents_MR(Regions_MR,Activities_MR,"Average residues (t)",
         "DM (t) - before management",ProductionSystems,ProductionConditions,"%Scenario%")
   = VCropResidues_Contents_MR.l(Regions_MR,Activities_MR,"Average residues (t)",
     "DM (t) - before management",ProductionSystems,ProductionConditions,"%Scenario%")
        /CropResidues_Contents_MR(Regions_MR,Activities_MR,"Average residues (t)",
          "DM (t) - before management",ProductionSystems,ProductionConditions,"%Scenario%");
```

```
        1.3) Derive other nutrient such as N and P2O5 contents of residues:
```

Multiplication of residue quantities with per ton contents delivers total nutrient quantities, etc., e.g. for nitrogen (and similarly for P_2O_5):

```
VCropResidues_Contents_MR.l(Regions_MR,Activities_MR,"Average residues (t)",
      "N (t) - before management",ProductionSystems,ProductionConditions,"%Scenario%")
   = VCropResidues_Quantity_MR.l(Regions_MR,Activities_MR,"Average residues (t)",
      ProductionSystems,ProductionConditions,"%Scenario%")
    *CropResidues_Contents_MR(Regions_MR,Activities_MR,"Average residues (t)",
      "N (t) - before management",ProductionSystems,ProductionConditions,"%Scenario%");
```

```
        1.4) Crop res management characteristics from total residues
```

Derive some further numbers based on total residue quantities, e.g. further characteristics such as management losses and emissions, etc., as covered in the set `CropResManagement_NotSystemShares`. As those characteristics may depend on the crop residue management system, this dimension is added by the share of residues managed in the different systems.

```
VCropResidues_Management_MR.l(Regions_MR,Activities_MR,"Average Residues (t)",
      CropResManagement_NotSystemShares,CropResManSystem,
             ProductionSystems,ProductionConditions,"%Scenario%")
   = VCropResidues_Quantity_MR.l(Regions_MR,Activities_MR,"Average residues (t)",
      ProductionSystems,ProductionConditions,"%Scenario%")
      *CropResidues_Management_MR(Regions_MR,Activities_MR,"Average Residues (t)",
         "Quantity share in CropResMan system",CropResManSystem,ProductionSystems,
         ProductionConditions,"%Scenario%")
           *CropResidues_Management_MR(Regions_MR,Activities_MR,"Average Residues (t)",
              CropResManagement_NotSystemShares,CropResManSystem,ProductionSystems,
                ProductionConditions,"%Scenario%");
```

And for forest residues (bioenergy), the N in forest bioenergy residues that can be applied as fertilizer is derived:

```
ActForest_OtherChar_MR.l(Regions_MR,"Forest","Forest res bioe N for areas (tN)",
      ProductionSystems,ProductionConditions,"%Scenario%")
   = VEnergyProduction_MR.l(Regions_MR,"Secondary energy","AllEnTypes",
         "Forest and crop residues","Bioe conversion Biomass input (tDM)","%Scenario%")
      *ActForest_OtherChar_MR(Regions_MR,"Forest",
           "Forest res bioe N for areas (tN/t DM res)",ProductionSystems,
              ProductionConditions,"%Scenario%");
```

To run these calculations, **no inputs need to be specified** in the scenario specifications (as all values are available via the initialization (cf. section 3.2.6); but they can be changed in the scenario specifications, if needed).

3.2.12 _V6_DataDerived_CropGrassNutrientRequirements

This file contains the code for an alternative approach to determine crop and grass nutrient requirements (than to use the data provided in the baseline), based on the nutrient contents in the output, accounting for crop residue output and N from N fixation. This is only assigned, though, in case no other, better data is available from the baseline or scenario specifications. Currently, however, there is no data read in in the baseline, thus all requirements are derived here (unless new data is added to improve the baseline or specified in scenario specifications). The table of contents is:

```
$ontext;
- 1) Crop and grass nutrient requirements derived
$offtext;
```

In more detail:

```
- 1) Crop and grass nutrient requirements derived
```

The N requirements per ton yield are derived as a proxy via total N in outputs (assuming a maximum of three outputs; for most cases, it is one only; as the reference is the first main output, the others are scaled proportionally to arrive at values per ton first output) and residues minus total N fixation with some correction factors; these correction factors still need to be backed by literature/improved. The requirements are only calculated if no value is available from the data (hence the condition "`$(NOT…`".). Thus, the N contents in the various outputs and the residues is calculated and then the total requirements are derived as follows (if a negative value results, this is set to zero in subsequent code not reported here).

These values are then later used to allocate total nutrients available in a region to the single crops. Hence it is not important whether this value exactly equals an agronomically sensible value for nutrient requirements or not. It is rather important that it correlates with such requirement as the relative values between crops govern how much of the total nutrient quantity is then applied to which crop (cf. section 3.2.17). Thus, this gross indicator for requirements is adequate in absence of better values.

```
ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,
       "N req - per ton yield based (tN)",ProductionSystems,ProductionConditions,
              "%Scenario%")
           $((NOT ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,
              "N req - per ton yield based (tN)",ProductionSystems,ProductionConditions,
                   "%Scenario%"))
              AND ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
                "MainOutput1 (t)",ProductionSystems,ProductionConditions,"%Scenario%"))
= ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"N in MainOutput1 (tN)",
       ProductionSystems,ProductionConditions,"%Scenario%")
   +ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"N in MainOutput2 (tN)",
       ProductionSystems,ProductionConditions,"%Scenario%")
         *ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"MainOutput2 (t)",
              ProductionSystems,ProductionConditions,"%Scenario%")
         /ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"MainOutput1 (t)",
              ProductionSystems,ProductionConditions,"%Scenario%")
   +ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"N in MainOutput3 (tN)",
       ProductionSystems,ProductionConditions,"%Scenario%")
         *ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"MainOutput3 (t)",
              ProductionSystems,ProductionConditions,"%Scenario%")
         /ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"MainOutput1 (t)",
              ProductionSystems,ProductionConditions,"%Scenario%")
   + 0.5*ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,
       "N in residues - per ton MainOutput1 (tN)",ProductionSystems,ProductionConditions,
              "%Scenario%")
   -0.75*ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,
       "N fixation per ton MainOutput1 (tN)",ProductionSystems,ProductionConditions,
              "%Scenario%");
```

After these calculations for nitrogen, the same is calculated for phosphorus (in P_2O_5).

To run these calculations, **no inputs need to be specified** in the scenario specifications (as all values are available via the initialization (cf. section 3.2.6); but they can be changed in the scenario specifications, if needed).

3.2.13 _V6_CoreModelEquations_NutrReqAndFeedSupply_DetailedFeedingRations

This file contains all the core model equations needed to derive animal nutrient requirements and feed supply - using detailed commodity level feeding rations. The table of contents is:

```
$ontext;
- 1) Nutrient requirements
- 2) Feeding rations, feed supply and nutrient contents
       2.1) Some general assignments for feeding rations
       2.2) Feed supply from DAQ - contents and aggregates
       2.3) Feed GE contents per DM
$offtext;
```

In more detail:

```
- 1) Nutrient requirements
```

First, the FeedME (metabolisable energy) requirements for dairy cows is derived, based on requirements for milk production (depending on the milk yield), maintenance (basic metabolism), walking and pregnancy, which are all themselves derived in specific equations (for details, see the code file), as also done for the baseline (see section 3.2.6).

```
ActAnimalsHead_OtherChar_MR(Regions_MR,"Cattle","Producing_Dairy_Cattle",
       "FeedME_Req_Total (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
     $ActAnimalsHead_OtherChar_MR(Regions_MR,"Cattle","Producing_Dairy_Cattle",
       "Liveweight (t)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_OtherChar_MR(Regions_MR,"Cattle","Producing_Dairy_Cattle",
       "FeedME_Req_MilkProd (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
+ ActAnimalsHead_OtherChar_MR(Regions_MR,"Cattle","Producing_Dairy_Cattle",
       "FeedME_Req_Maintenance (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
+ ActAnimalsHead_OtherChar_MR(Regions_MR,"Cattle","Producing_Dairy_Cattle",
       "FeedME_Req_Walking (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
+ ActAnimalsHead_OtherChar_MR(Regions_MR,"Cattle","Producing_Dairy_Cattle",
       "FeedME_Req_Pregnancy (MJ)",ProductionSystems,ProductionConditions,"%Scenario%");
```

The $-condition on the presence of liveweight values is needed to avoid that the requirement is assigned also if one of the liveweight-related terms is missing (this likely being the case, e.g. when there are no animals in a country).

Then feed crude protein (FeedXP) requirements are derived for all animals, from FeedME requirements, by means of a proportionality factor (quite a gross approach), in case there is no better data available (hence the condition “`$(NOT…`”.):

```
ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "FeedXP_Req_Total (t)",ProductionSystems,ProductionConditions,"%Scenario%")
    $(ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "XPperME_InFeedReq (gXP/MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
     AND (NOT ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "FeedXP_Req_Total (t)",ProductionSystems,ProductionConditions,"%Scenario%")))
= ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "FeedME_Req_Total (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
     *ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "XPperME_InFeedReq (gXP/MJ)",ProductionSystems,ProductionConditions,
       "%Scenario%")/1000000;
```

And after this, gross energy (FeedGE) requirements are derived for all animals, from FeedME requirements, based on equations from (IPCC 2006):

```
ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "FeedGE_Req_Total (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
    $((ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Digestibility of Feed (%)",ProductionSystems,ProductionConditions,"%Scenario%")/100
```

```
        - ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
                "UE_per_GE (share)",ProductionSystems,ProductionConditions,"%Scenario%"))
     AND NOT ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "FeedGE_Req_Total (MJ)",ProductionSystems,ProductionConditions,"%Scenario%"))
= ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "FeedME_Req_Total (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
     /(ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Digestibility of Feed (%)",ProductionSystems,ProductionConditions,"%Scenario%")/100
        - ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
                "UE_per_GE (share)",ProductionSystems,ProductionConditions,"%Scenario%"));
```

IMPORTANT: if the equations to derive the dairy cow FeedME requirements are changed here, then they also need to be changed identically in section 3.2.6, where the baseline of this is derived.

```
- 2) Feeding rations, feed supply and nutrient contents
        2.1) Some general assignments for feeding rations
```

The following parameters for feeding rations are assigned: commodity specific per head and per ton feed values, specified per animal, to allow for the calculations with commodity- and animal-specific detailed feeding rations.

IMPORTANT: The starting point to derive these values are the following two parameters, that either have to be taken from default baseline values or set in the scenario specifications:
1) DM quantity shares for each commodity in per head commodity-specific feeding
2) rationsTotal GE feed requirement per head from ALL feed

```
FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,
        AnimalTypeInHerd,Commodities_Feed,
                "Quantity share in DM (share)",
                        ProductionSystems,ProductionConditions,"%Scenario%")

ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
                "FeedGE_Req_Total (MJ)",
                        ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,Commodities_Feed,
        "Quantity share in DM (share)",
                ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRations_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,Commodities_Feed,
        "GE per ton DM feeding ration (MJ/t DM)",
                ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,Commodities_Feed,
        "GE share in feeding ration (share in MJ)",
                ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,Commodities_Feed,
        "FeedQuant_Req (t)",
                ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,Commodities_Feed,
        "FeedQuant_Req DM (t)",
                ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,Commodities_Feed,
        "FeedGE_Req (MJ)",
                ProductionSystems,ProductionConditions,"%Scenario%")
```

Furthermore, the total fresh and dry matter feed quantities per head are derived:

```
FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "AggregateFeedGrass_Commodity","FeedQuant_Req (t)",
                ProductionSystems,ProductionConditions,"%Scenario%")
```

```
FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "AggregateFeedGrass_Commodity","FeedQuant_Req DM (t)",
                ProductionSystems,ProductionConditions,"%Scenario%")
```

```
        2.2) Feed supply from DAQ - contents and aggregates
```

Total Feed supply for each commodity is already derived in _V6_DataDerived_CropProductionTotalsAndDAQ - derive here various aggregates and content values of this.
This is aggregates of VCommod_Feed_MR for various feed groups, using three matching sets:
1) First, for the feed groups with concentrates such that all concentrates are put together (groups: concentrates, forage, grass (no entries for residues)
2) Then, with concentrates such that main and byprod concentrates are kept in separate groups (groups: concentratesMainProd, ConcentratesByProd, forage, grass (no entries for residues)
3) then do it for more detailed groups: this covers all, and various crop groups, and also all without grass/forage - but no differentiation main/byprod, as this is not available in the data (e.g. Wheat and products" contains both)

Then, dry matter and GE, ME and XP quantities are derived per commodity
Then these are again aggregated to the feed groups as done above for the fresh matter quantities.

For details, see the code.

2.3) Feed GE and XP contents per DM
*derive Feed GE and XP contents per DM for the feed commodities, feed groups and total feed, for details, see the code.

3.2.14 _V6_CoreModelEquations_DeriveAnimalNumbAndProd_DetailedFeedingRations

This file contains all the core model equations for deriving animal numbers from feed supply and feeding rations and for deriving animal production, imports/exports and domestically available quantities - this version uses DETAILED FEEDING RATIONS and not the feed group aggregates

The calculations are done iteratively:
- First, ruminants (animals with grass in the feeding ration) are derived from grass availability; then their demand for concentrates is derived and subtracted from total concentrates supply.
- At this stage, concentrates are purely plant based - derived from the plant production DAQ calculations
- Then, the various commodities that are produced from these animals are calculates, including imports, exports, DAQ - and their utilisation.
- The remaining concentrates quantity is then augmented by the feed utilisation of these animal products, this total is then used to derive the number of non-grass-fed animals.
- Then the commodity quantities from these non-grass-fed animals are derived (including import, export, DAQ) - and utilisations - part of this is again feed.
- This additional feed is then used to feed some additional non-grass-fed animals, and these result in a slightly increased number of commodities, then the iteration stops
(but could be continued- and coded as a genuine loop - if needed; it is not yet looped as in this first two iterations the code looks slightly different, cf. below)).

The table of contents is:

```
$ontext;
```

```
- 1) Derive numbers of animals with grass in the feeding rations
- 2) Derive primary product production from the animals with grass in the feeding rations
- 3) Derive commodity production from the animals with grass in the feeding rations
        3.1) Production of primary products as commodities
        3.2) Production, Import, export and DAQ values for commodities that are not primary
             products as commodities
               3.2.1) Commodities that directly correspond to primary products from
                         activities but are mostly named differently
               3.2.2) Derived grass-fed animal commodities that are directly derived from
                         primary commodities for which trade and production data and thus
                         DAQ are available
- 4) Fish and seafood:
- 5) Utilization of DAQ and some other values for commodities from animals with grass in the
       feeding ration
- 6) Derive numbers of animals WITHOUT grass in the feeding rations - First iteration
        6.1) Concentrate feed supply to the animals WITHOUT grass in the feeding rations
        6.2) Derive suckled and producing animals WITHOUT grass in the feeding rations
- 7) Derive number of beehives and honey production
- 8) Derive primary product production from the animals WITHOUT grass in the feeding rations –
       first iteration
- 9) Derive commodity production from the animals WITHOUT grass in the feeding rations - first
       iteration
        9.1) Production of primary products as commodities
        9.2) Production, Import, export and DAQ values for commodities that are not primary
             products as commodities
               9.2.1) Commodities that directly correspond to primary products from
                         activities but are mostly named differently - non-grass-fed
                         animals or all animals
               9.2.2) Derived non-grass-fed animal commodities that are directly derived
                         from primary commodities for which trade and production data and
                         thus DAQ are available
- 10) Utilization of DAQ and some other values for commodities from animals WITHOUT grass in
       the feeding ration - first iteration
- 11) Adjust numbers of animals WITHOUT grass in the feeding rations: - Second iteration
- 12) Derive primary product production from the animals WITHOUT grass in the feeding rations
       - second iteration
- 13) Adjust commodity production from the animals WITHOUT grass in the feeding rations –
       Second iteration
        13.1) Production of primary products as commodities
        13.2) Production, Import, export and DAQ values for commodities that are not primary
             products as commodities
               13.2.1) Commodities that directly correspond to primary products from
                    activities but are mostly named differently - non-grass-fed animals or
                    all animals
               13.2.2) Derived non-grass-fed animal commodities that are directly derived
                    from primary commodities for which trade and production data and thus
                    DAQ are available
- 14) Utilization of DAQ and some other values for commodities from animals WITHOUT grass in
       the feeding ration - second iteration
- 15) Detailed Feeding Rations
- 16) Derive feed requirements per APU
$offtext;
```

For more details, see the code file.

Here, we disply the following formulae only, indicating how the number of non-grass fed animals is derived:

```
$ontext;
The proportionality between different animals is then used as follows to derive unknown parameters in the equations to derive animal numbers:

                                                CerealSupplyLeftFromRumi * ShareOfCerealDemandOfTheseAnimalsInCerealsLeft
NumberOfAnimals(From Cereals Left from Ruminants) = --------------------------------------------------------------------
                                                SupplyDemandRatioOfCereals * (TotalperHeadCerealRequirements)

In this, the second factor in the numerator is not known.
But the following holds for this second factor:

                                                ShareOfCerealDemandOfTheseAnimalsInALLCereals
ShareOfCerealDemandOfTheseAnimalsInCerealsLeft =     ---------------------------------------------------------
                                                SUM_AnimalsLeft(ShareOfCerealsDemandOfTheseAnimalsInALLCereals)

And inserting this above results in the following:

                                                CerealSupplyLeftFromRumi * ShareOfCerealDemandOfTheseAnimalsInALLCereals
NumberOfAnimals(From Cereals Left from Ruminants) = ------------------------------------------------------------------------------------------
                                                SupplyDemandRatioOfCereals * (TotalperHeadCerealRequirements) * SUM_AnimalsLeft(ShareOfCerealDemandOfTheseAnimalsInALLCereals)

thereby, supply/demand ration and ShareOfCerealDemandOfTheseAnimalsInALLCereals are taken from the baseline

$offtext;
```

To run all the animal number calculations, the following inputs **may/likely need to be specified** in the scenario specifications, if no default values shall be used (all other values are available via the initialization (cf. section 3.2.6); but also they can be changed in the scenario specifications, if needed). This is so, as changes in feeding rations and feed use is a core aspect of many scenarios.

- share of the specific animal type in total quantity feed requirements of all animals, for all grass commodities aggregated and for all cereal commodities aggregated. The element "Quantity share in total feed req of all animals" is currently used and available only for the aggregate grass and cereal commodities, not for single commodities, hence it has to be directly specified at this level of aggregation in the scenario specifications, if the baseline values should be changed.

Importantly, changes in this have to be consistent over all animals and both grass and cereal aggregates. Cereals play a special role as key concentrate feed that is then used to derive the non-grass-fed animals.

```
FeedingRations_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "AggregateFeedGrass_Commodity",
              "Quantity share in total feed req of all animals",
                     ProductionSystems,ProductionConditions,"%Scenario%")

FeedingRations_OtherChar_MR(Regions_MR,Livestock_MR,AnimalTypeInHerd,
       "AggregateFeedCereals_Commodity",
              "Quantity share in total feed req of all animals",
                     ProductionSystems,ProductionConditions,"%Scenario%")
```

Central are also the DM requirement per head of the specific animal type, for all grass commodities aggregated and for all cereal commodities aggregated – but they are derived by aggregating the respective single commodity values that are already specified (see 3.2.13) over the respective commodity groups:

```
FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "AggregateFeedGrass_Commodity",
              "FeedQuant_Req DM (t)",
                     ProductionSystems,ProductionConditions,"%Scenario%")
FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "AggregateFeedCereals_Commodity",
              "FeedQuant_Req DM (t)",
                     ProductionSystems,ProductionConditions,"%Scenario%")
```

- DAQ of fish and seafood commodities, as these are not yet internally determined and are just set equal to the baseline value per default, if not replaced by other values

```
VCommod_Quantity_MR.l(Regions_MR,
       CommoditiesDAQ_TradeFromToWorld_FishSeafood,
              ProductionSystems,ProductionConditions,"%Scenario%")
```

- Number of beehives, as these are internally determined by some gross proportionality to pollination services demand, depending on the cultures that require such – but this default may often be decided to be replaced by other values via scenario specifications

```
VActAnimalsAPU_QuantityActUnits_MR.l(Regions,Activities,AnimalTypeInHerd,
       ProductionSystems,ProductionConditions,"%Scenario%")
```

3.2.15 _V6_CoreModelEquations_ManureExcretionAndManagement

This file contains all the equations for calculating manure excretion, manure management and related emissions, based on (IPCC 2006). The table of contents is:

```
$ontext;
- 1) Manure excretion
- 2) Manure management
         2.1) Some general calculations
         2.2) Methane emissions
         2.3) N2O emissions
                 2.3.1) Direct emissions
                 2.3.2) Indirect Emissions
                   2.3.2 A) indirect emissions from volatilisation:
                   2.3.2 B) indirect emissions from leach/runoff:
                 2.3.3) Total N and GHG emissions and losses from manure management
         2.4) P emissions
         2.5) N and P available for fertilization
$offtext;
```

In more detail:

```
- 1) Manure excretion
```

derive VS - Volatile solidesexcretion rates - are determined as follows:

VS = [GE*(1 - DE%/100) + UE*GE] * [(1 - ASH)/GE_per_DM]

unit: t VS in DM/animal/year

Where:

VS = volatile solid excretion per year on a dry-organic matter basis, kg VS year-1

GE = gross energy intake, MJ year-1

DE% = digestibility of the feed in percent (e.g. 60%)

(UE*GE) = urinary energy expressed as fraction of GE times GE. Typically 0.04GE can be considered urinary energy excretion by most ruminants (reduce to 0.02 for ruminants fed with 85% or more grain in the diet or for swine). Use country-specific values where available.

ASH = the ash content of manure calculated as a fraction of the dry matter feed intake (e.g., 0.08 for cattle). Use country-specific values where available.

GE_per_DM = conversion factor for dietary GE per kg of dry matter (MJ kg-1). This value is relatively constant across a wide range of forage and grain-based feeds commonly consumed by livestock. IPCC Default: 18.45 - we use the values derived from feed supply, which are largely ok, it seems.

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "VS DM in manure (tVS)",ProductionSystems,ProductionConditions,"%Scenario%")
     $Commod_Contents_MR(Regions_MR,"AggregateFeedTotal_Commodity",
       "FeedGE in DM (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "FeedGE_Req_Total (MJ)",ProductionSystems,ProductionConditions,"%Scenario%")
     *(1 - ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Digestibility of Feed (%)",ProductionSystems,ProductionConditions,"%Scenario%")/100
      + ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "UE_per_GE (share)",ProductionSystems,ProductionConditions,"%Scenario%"))
             *(1 - Manure_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
```

```
        "Ash content in feed DM (share)",ProductionSystems,ProductionConditions,
        "%Scenario%"))
   /Commod_Contents_MR(Regions_MR,"AggregateFeedTotal_Commodity",
        "FeedGE in DM (MJ)",ProductionSystems,ProductionConditions,"%Scenario%");
```

Then derive manure total solids (TS). Assume total solids to be volatile solids plus ash, then being equal to total dry matter. Unit: t TS/animal/year:

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "VS DM in manure (tVS)",ProductionSystems,ProductionConditions,"%Scenario%")
    + ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "FeedDM_Req_Total (t DM)",ProductionSystems,ProductionConditions,"%Scenario%")
      *Manure_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Ash content in feed DM (share)",ProductionSystems,ProductionConditions,"%Scenario%");
```

```
- 2) Manure management
         2.1) Some general calculations
```

This contains some general calculations to derive N and P excretion per head (from the values per ton liveweight), and N and P contents of manure totale and volatile solids. For details, see the code file.

```
         2.2) Methane emissions
```

Methane emissions from manure management are calculated based on (IPCC 2006), Tier 2 approach, Volume 4, chapter 10, equations 10.23 ff.

Methane emissions are calculated in the unit: m^3 CH_4 / kg manure excreted

EF(T) = (VS(T) * 365) * (Bo(T) * 0.67 kg/m3 * Sum(S,k)MCF(S,k)/100*MS(T,S,k))

where

EF(T) = annual CH_4 emission factor for livestock category T, kg CH_4 animal^{-1} yr^{-1}

VS(T) = daily volatile solid excreted for livestock category T, kg dry matter animal^{-1} day^{-1}

365 = basis for calculating annual VS production, days yr^{-1}

Bo(T) = maximum methane producing capacity for manure produced by livestock category T, m^3 CH_4 kg^{-1} of VS excreted

0.67 = conversion factor of m^3 CH_4 to kilograms CH_4

MCF(S,k) = methane conversion factors for each manure management system S by climate region k, %

MS(T,S,k) = fraction of livestock category T's manure handled using manure management system S in climate region k, dimensionless

thereby, VS - Volatile solids excretion rates - are determined as follows:

VS = [GE*(1 - DE%/100) + UE*GE] * [(1 - ASH)/GE_per_DM]

Where:

VS = volatile solid excretion per day on a dry-organic matter basis, kg VS day^{-1}

GE = gross energy intake, MJ day^{-1}

DE% = digestibility of the feed in percent (e.g. 60%): thus choose 0.6

(UE*GE) = urinary energy expressed as fraction of GE. Typically 0.04GE can be considered urinary energy excretion by most ruminants (reduce to 0.02 for ruminants fed with 85% or more grain in the diet or for swine). Use country-specific values where available.

ASH = the ash content of manure calculated as a fraction of the dry matter feed intake (e.g., 0.08 for cattle). Use country-specific values where available.

GE_per_DM = conversion factor for dietary GE per kg of dry matter (MJ kg-1). This value is relatively constant across a wide range of forage and grain-based feeds commonly consumed by livestock. IPCC Default: 18.45 - we use the values derived from feed supply, which are largely ok, it seems.

The following data is used:
Bo(T): tables 10A-4 to 10A-9
MCF(S,k): table 10.17

Most of these parameters are already defined in the modules for reading data on manure excretion and manure management, etc.

Now do the calculations:
*EF(T) = (VS(T)) * (Bo(T) * 0.67 kg/m3 * Sum(S,k)MCF(S,k)/100*MS(T,S,k))
That is the original IPCC-formula - but for per animal values, we refrain from summing over climate, as one animal is usually located in one climate only.
In the IPCC-formula, there is a factor 365, as it is per day value for VS - but we have annual values, thus no 365 needed (cf. the baseline data code on manure excretion, section 3.1.51)
VS is: `ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"VS DM in manure (tVS)",ProductionSystems,ProductionConditions,"%Scenario%")`
unit: t CO_2eq/head/year;
division by 100 at MCF in the formula above for transforming percentages to shares/fractions
Bo*0.67kg/m^3 is kgCH_4/kg VS, thus equal to tCH_4/tVS

Now derive the CH_4-emissions (in tCH_4 per head):

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man CH4 (tCH4)" ,ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "VS DM in manure (tVS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystem,Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Bo: max. CH4 prod. cap. (m3CH4/kgVS)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")
     *0.67*Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "MCF: CH4 conversion factor (%)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")/100
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Quantity share in ManureMan system",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%"));
```

and derive the CH_4-emissions in tCO_2eq per head:

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man CH4 (tCO2e)" ,ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man CH4 (tCH4)" ,ProductionSystems,ProductionConditions,"%Scenario%")
     *GWP_GTP_SOLm_MR("CH4","%Scenario%");
```

Now derive the CH_4-emissions per ton of manure TS DM in one specific manure management system:
Emissions/t manure (Manman syst) = (Emissions/head (If all manure would be managed in ONE manman syst) / (Manure/head); unit: t CO_2eq / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man CH4 (tCH4)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
       ProductionConditions,"%Scenario%")
     $ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "VS DM in manure (tVS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Bo: max. CH4 prod. cap. (m3CH4/kgVS)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")
      *0.67*Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "MCF: CH4 conversion factor (%)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")/100
     /ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%");
```

And also in tCO_2eq per head:

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man CH4 (tCO2e)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man CH4 (tCH4)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
     *GWP_GTP_SOLm_MR("CH4","%Scenario%");
```

```
          2.3) N2O emissions
                2.3.1) Direct emissions
```

Direct N_2O-emissions from manure management are calculated according to (IPCC 2006), volume 10, chapter 4, equation 10.25:

N2O_direct_perHead = Nex(T) * MS(T,S) * EF3(S) * 44/28

unit: tons CO_2eq/head/year

Where:

N2O_direct_perHead = direct N2O emissions from Manure Management in the country, per head of animal, kg N_2O yr^{-1}

Nex(T) = annual average N excretion per head of species/category T in the country, kg N $animal^{-1}$ yr^{-1}

MS(T,S) = fraction of total annual nitrogen excretion for each livestock species/category T that is managed in manure management system S in the country, dimensionless

EF3(S) = emission factor for direct N_2O emissions from manure management system S in the country, kgN_2O-N/kg N in manure management system S (values from Table 10.21)

S = manure management system

T = species/category of livestock

44/28 = conversion of N_2O-N emissions to N_2O emissions

Direct N_2O emissions from manure management in tN_2O/head:

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
    *sum(ManureManSystem,
        Manure_Management_MR(Regions_MR,Activities_MR,
          AnimalTypeInHerd,"Manure man N2O dir (tN2O-N/tN)",ManureManSystem,
          "AllAndAverageTemp",ProductionSystems,ProductionConditions,"%Scenario%")
         *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
          "Quantity share in ManureMan system",ManureManSystem,"AllAndAverageTemp",
          ProductionSystems,ProductionConditions,"%Scenario%"))
     *44/28;
```

Converted to tCO_2eq/head:

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
     *GWP_GTP_SOLm_MR("N2O","%Scenario%");
```

Derive the N quantity lost through these emissions (divide the N_2O emissions by the N_2O-N to N_2O-factor 44/28:

unit: tons N/head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N in N2O dir (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
     /44*28;
```

And derive the direct N losses via N_2O-emissions per ton manure TS DM:

*tN Losses/t manure (Manman syst) = (Losses/head (If all manure would be managed in ONE manman syst) / (Manure/head) unit: t N / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
"Manure man N in N2O dir (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
     $ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tN2O-N/tN)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
    /ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%");
```

Derive the corresponding N_2O emissions (in tCO_2eq) per ton manure:

unit: t CO_2eq / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tCO2e)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N in N2O dir (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
      *44/28*GWP_GTP_SOLm_MR("N2O","%Scenario%");
```

```
                 2.3.2) Indirect Emissions
                   2.3.2 A) indirect emissions from volatilisation:
```

Indirect N_2O emissions are calculated according to eq. 10.26 and 10.27 in (IPCC 2006):

N_2O_indirect_perHead_Volat = Nex(T) * MS(T,S) * (FracGasMS/100) * EF4(S) * 44/28

Where:

FracGasMS = percent of managed manure nitrogen for livestock category T that volatilises as NH_3 and NO_x in the manure management system S, unit: percentage; values are from table 22, (IPCC 2006), vol4, chapter 10.

EF4 = emission factor for N_2O emissions from atmospheric deposition of nitrogen on soils and water surfaces, kg N_2O-N (kg NH_3-N + NO_x-N volatilised)$^{-1}$; the default value is 0.01 kg N_2O-N (kg NH_3-N + NO_x-N volatilised)$^{-1}$, from (IPCC 2006) Chapter 11, Table 11.3

unit: share (i.e. percentage/100)

Now do these calculations for indirect volatilisation N_2O emissions:

*unit: t N_2O/head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystem,
     Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")/100
    *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O-N from N volat (tN/tN volat)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
    *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Quantity share in
        ManureMan system",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%"))
      *44/28;
```

Now convert to t CO2eq/head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
      *GWP_GTP_SOLm_MR("N2O","%Scenario%");
```

Then calculate the related N quantity lost through volatilization:

unit: t N /head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
      *sum(ManureManSystem,
      Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")/100
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Quantity share in
        ManureMan system",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%"));
```

Derive NH_3-emissions as a specific part of the volatilization:

unit: t NH_3/head/year (17/14 converts N to NH_3)

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man NH3 (tNH3)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
      *sum(ManureManSystem,
        Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
                "Manure man NH3-N (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
                ProductionSystems,ProductionConditions,"%Scenario%")/100
       *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
                "Quantity share in ManureMan system",ManureManSystem,"AllAndAverageTemp",
                ProductionSystems,ProductionConditions,"%Scenario%"))
           *17/14;
```

Now convert the per head values to per ton manure values: derive first N lost through volatilisation per ton manure TS DM:

Emissions/t manure (Manman syst) = (Emissions/head (If all manure would be managed in ONE manman syst) / (Manure/head)

unit t N / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
    $ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
    *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")/100
    /ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%");
```

and corresponding indirect N_2O emissions from volatilisation:

unit: t N_2O / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tN2O)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
    *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O-N from N volat (tN/tN volat)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
      *44/28;
```

And corresponding GHG emissions, i.e. converted to t CO_2eq / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tCO2e)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tN2O)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
    *GWP_GTP_SOLm_MR("N2O","%Scenario%");
```

and also NH_3 emissions as specific part of the volatilization:

unit t NH_3 / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man NH3 (tNH3)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
       ProductionConditions,"%Scenario%")
     $ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man NH3-N (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")/100
     /ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *17/14;
```

2.3.2 B) indirect emissions from leach/runoff:

This is done with equations 10.28 and 10.29 from (IPCC 2006):

N_2O_indirect_perHead_Leach = Nex(T) * MS(T,S) * (FracLeachMS/100) * EF5(S) * 44/28

with

FracleachMS = percent of managed manure nitrogen losses for livestock category T due to runoff and leaching during solid and liquid storage of manure (typical range 1-20%)

EF5 = emission factor (share - i.e. %/100) for N_2O emissions from nitrogen leaching and runoff, kg N_2O-N/kg N leached and runoff (default value 0.0075 kg N_2O-N (kg N leaching/runoff)$^{-1}$, given in Chapter 11, Table 11.3

unit: t N2O/head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N2O leach (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
    *sum(ManureManSystem,
       Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N leach (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")/100
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N2O-N from N leach (tN/tN leach)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Quantity share in ManureMan system",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%"))
      *44/28;
```

Convert to t CO2eq/head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N2O leach (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N2O leach (tN2O)",ProductionSystems,ProductionConditions,"%Scenario%")
     *GWP_GTP_SOLm_MR("N2O","%Scenario%");
```

and calculate the related N quantity:

unit: t N /head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N leach (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
    *sum(ManureManSystem,
       Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure man N leach (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")/100
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Quantity share in ManureMan system",ManureManSystem,"AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%"));
```

Now derive N2O indirect leach per ton manure TS DM:

E/t manure (Manman syst) = (E/head (If all manure would be managed in ONE manman syst) / (Manure/head)

unit t N / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N leach (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
     $ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in manure (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N leach (% of N in manure)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")/100
     /ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%");
```

Derive the corresponding N_2O emissions:

unit: t N_2O / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O leach (tN2O)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N leach (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
    *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O-N from N leach (tN/tN leach)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
      *44/28;
```

Derive he corresponding GHG emissions:

unit: t CO_2eq / t manure TS DM

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O leach (tCO2e)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O leach (tN2O)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
      *GWP_GTP_SOLm_MR("N2O","%Scenario%");
```

```
                  2.3.3) Total N and GHG emissions and losses from manure management
```

Now sum the N emissions and losses via direct and indirect paths to get total N losses from manure management:

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O total (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
   + ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O volat (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
   + ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O leach (tCO2e)",ProductionSystems,ProductionConditions,"%Scenario%");

ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N total loss (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N in N2O dir (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
   + ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
   + ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N leach (tN)",ProductionSystems,ProductionConditions,"%Scenario%");

Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O total (tCO2e)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O dir (tCO2e)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
   + Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
```

```
        "Manure man N2O volat (tCO2e)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
   + Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N2O leach (tCO2e)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%");

Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N total loss (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N in N2O dir (tN)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
   + Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N volat (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
   + Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man N leach (tN)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%");
```

***IMPORTANT: Add N_2 losses from denitrification explicitly to track N_2 losses from denitrification for the N balances and thus to get a better grasp for the N surplus!!**

2.4) P emissions

Unit: tP_2O_5 lost/t manure TS DM per year

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man P total loss (tP2O5)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")
= Manure_Contents_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "P in TS DM (tP2O5/t TS DM)",ProductionSystems,ProductionConditions,"%Scenario%")
     *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man P loss as % of P in manure (%)",ManureManSystem,"AllAndAverageTemp",
        ProductionSystems,ProductionConditions,"%Scenario%")/100;
```

*derive per animal head values:

*unit: t P_2O_5 /head/year

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure man P total loss (tP2O5)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "P in manure (tP2O5)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystem,
        Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
               "Manure man P loss as % of P in manure (%)",ManureManSystem,
               "AllAndAverageTemp",ProductionSystems,ProductionConditions,"%Scenario%")/100
       *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
               "Quantity share in ManureMan system",ManureManSystem,"AllAndAverageTemp",
               ProductionSystems,ProductionConditions,"%Scenario%"));
```

2.5) N and P available for fertilization

Here, the reminder of N and P in manure, after subtracting losses, is calculated. This is the amount that is in principle available for fertilization.

```
Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure N for areas (tN)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Contents_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "N in TS DM (tN/t TS DM)",ProductionSystems,ProductionConditions,"%Scenario%")
     - Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
               "Manure man N total loss (tN)",ManureManSystem,"AllAndAverageTemp",
               ProductionSystems,ProductionConditions,"%Scenario%");

Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "Manure P for areas (tP2O5)",ManureManSystem,"AllAndAverageTemp",ProductionSystems,
        ProductionConditions,"%Scenario%")
= Manure_Contents_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
        "P in TS DM (tP2O5/t TS DM)",ProductionSystems,ProductionConditions,"%Scenario%")
     - Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
               "Manure man P total loss (tP2O5)",ManureManSystem,"AllAndAverageTemp",
```

```
            ProductionSystems,ProductionConditions,"%Scenario%");
```

Subsequently, negative values are set equal zero (see code file for details).
Then convert to per animal head values, and differentiate between N and P for crop and grassland areas:

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure N for crop areas (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystemCropland,
       Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Manure N for areas (tN)",ManureManSystemCropland,"AllAndAverageTemp",
              ProductionSystems,ProductionConditions,"%Scenario%")
       *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Quantity share in ManureMan system",ManureManSystemCropland,
              "AllAndAverageTemp",ProductionSystems,ProductionConditions,"%Scenario%"));

ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure N for grass areas (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystemGrassland,
       Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Manure N for areas (tN)",ManureManSystemGrassland,"AllAndAverageTemp",
              ProductionSystems,ProductionConditions,"%Scenario%")
       *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Quantity share in ManureMan system",ManureManSystemGrassland,
              "AllAndAverageTemp",ProductionSystems,ProductionConditions,"%Scenario%"));

ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure P for crop areas (tP2O5)",ProductionSystems,ProductionConditions,"%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystemCropland,
       Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Manure P for areas (tP2O5)",ManureManSystemCropland,"AllAndAverageTemp",
              ProductionSystems,ProductionConditions,"%Scenario%")
        *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Quantity share in ManureMan system",ManureManSystemCropland,
              "AllAndAverageTemp",ProductionSystems,ProductionConditions,"%Scenario%"));

ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Manure P for grass areas (tP2O5)",ProductionSystems,ProductionConditions,
       "%Scenario%")
= ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "TS DM in manure (tTS)",ProductionSystems,ProductionConditions,"%Scenario%")
     *sum(ManureManSystemGrassland,
       Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Manure P for areas (tP2O5)",ManureManSystemGrassland,"AllAndAverageTemp",
              ProductionSystems,ProductionConditions,"%Scenario%")
    *Manure_Management_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
              "Quantity share in ManureMan system",ManureManSystemGrassland,
              "AllAndAverageTemp",ProductionSystems,ProductionConditions,"%Scenario%"));
```

To run these calculations, **no inputs need to be specified** in the scenario specifications (as all values are available via the initialization (cf. section 3.2.6); but they can be changed in the scenario specifications, if needed).

3.2.16 _V6_CoreModelEquations_EntericFerm_DetailedFeedingRations

This file contains the equations for calculating enteric fermentation emissions for the DETAILED feeding rations, based on (IPCC 2006). The table of contents is:

```
$ontext;
- 1) Enteric fermentation
```

```
$offtext;
```

In more detail:

```
- 1) Enteric fermentation
```

This calculates enteric fermentation as a function of GE uptake, (IPCC 2006), chapter 10, tier 2, equation 10.21:

EF = (GE * Ym/100 * 365) / 55.65

with

EF = emission factor, kg CH_4 head^{-1} yr^{-1}

GE = gross energy intake, MJ head^{-1} year^{-1} (in the IPCC 2006 documents it is per day, but here in the code it is already per year, as GE is per year: thus, NO factor 365 is needed, but a division by 1000 is needed, as we want tons per year, and not kg per year.

Ym = methane conversion factor, per cent of gross energy in feed converted to methane coded as `FeedingRations_OtherChar_MR(Regions,Activities,AnimalTypeInHerd,Commodities,"Percentage GE in feed converted to enteric CH4",ProductionSystems,ProductionConditions,"Baseline")`.

The factor 55.65 (MJ/kg CH_4) is the energy content of methane.

Ym is taken from tables 10.12 und 10.13; the values are basically 6.5% for all cases besides >90% concentrates-fed cattle: there it is 3%. And lambs from sheep have 4.5% (< 1 year). Thus assume a linear relation with the share of concentrates, and disregard the special value for lamb for now, as we do not differentiate the different roles in the herd for sheep.

```
ActAnimalsHead_Outputs_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
       "Enteric ferment (t CO2e)",ProductionSystems,ProductionConditions,"%Scenario%")
         = sum(Commodities_Feed,FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,
              AnimalTypeInHerd,Commodities_Feed,"FeedGE_Req (MJ)",ProductionSystems,
                     ProductionConditions,"%Scenario%")
               *FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,
                     Commodities_Feed,"Percentage GE in feed converted to enteric CH4",
                     ProductionSystems,ProductionConditions,"%Scenario%"))
               *GWP_GTP_SOLm_MR("CH4","%Scenario%")
                     /100/55.65/1000;
```

To run these calculations, **no inputs need to be specified** in the scenario specifications (as all values are available via the initialization (cf. section 3.2.6); but they can be changed in the scenario specifications, if needed).

3.2.17 _V6_CoreModelEquations_FertilizerApplication

This file contains the equations for fertilizer application. The table of contents is:

```
$ontext;
- 1) Total fertilizer supply
         1.1) Manure
         1.2) Crop residues
         1.3) Mineral fertilizers
- 2) Relative nutrient requirements for the different crops and grasses
- 3) Fertilizer allocation to crops and grassland
$offtext;
```

In more detail:

```
- 1) Total fertilizer supply
         1.1) Manure
```

Derive total manure N available to be put on the field (crop and grass), per geographic unit. This is based on per animal head values for manure N for areas, multiplied with animal numbers and summed over all animals:

```
VManure_Management_MR.l(Regions_MR,"All Animals","AllAndAverageTypes",
       "Manure N for crop areas (tN)","AllManManSystems","AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")
= sum((Livestock_MR,AnimalTypeInHerd_MR),
       ActAnimalsHead_Outputs_MR(Regions_MR,Livestock_MR,AnimalTypeInHerd_MR,
```

```
        "Manure N for crop areas (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
    *VActAnimalsHead_QuantityActUnits_MR.l(Regions_MR,Livestock_MR,AnimalTypeInHerd_MR,
             ProductionSystems,ProductionConditions,"%Scenario%"));

VManure_Management_MR.l(Regions_MR,"All Animals","AllAndAverageTypes",
       "Manure N for grass areas (tN)","AllManManSystems","AllAndAverageTemp",
       ProductionSystems,ProductionConditions,"%Scenario%")
= sum((Livestock_MR,AnimalTypeInHerd_MR),
       ActAnimalsHead_Outputs_MR(Regions_MR,Livestock_MR,AnimalTypeInHerd_MR,
         "Manure N for grass areas (tN)",ProductionSystems,ProductionConditions,"%Scenario%")
    *VActAnimalsHead_QuantityActUnits_MR.l(Regions_MR,Livestock_MR,AnimalTypeInHerd_MR,
             ProductionSystems,ProductionConditions,"%Scenario%"));
```

Similarly, derive the total manure P available to be put on the field (crop and grass), per geographic unit (for details, see the code file).

1.2) Crop residues

Derive the total crop residue N and P available to be put on the field, per geographic unit. Sum over all crops and crop residue management systems (also including “left on field” - this is also applied to other crops due to other crops standing next season, etc. – on average, such a sum is thus ok):

```
VCropResidues_Management_MR.l(Regions_MR,"All Crops","Average Residues (t)",
       "Crop res N for areas (tN)","AllAndAverageCropResManSystem",ProductionSystems,
       ProductionConditions,"%Scenario%")
= sum((Crops_MR,CropResManSystemCropland),
       VCropResidues_Management_MR.l(Regions_MR,Crops_MR,"Average Residues (t)",
              "Crop res N for areas (tN)",CropResManSystemCropland,ProductionSystems,
              ProductionConditions,"%Scenario%"));
```

Similarly, derive the total crop residue P available to be put on the field, per geographic unit (for details, see the code file).

For grass, the residues are the difference between yield and harvest/grazing, thus they remain on the land, thus no sum over different grassland, etc. and no removal to cropland occurs, thus just use the following values:

```
VCropResidues_Management_MR.l(Regions_MR,CoreGrassActivities,"Average Residues (t)",
       "Crop res N for areas (tN)","AllAndAverageCropResManSystem",ProductionSystems,
       ProductionConditions,"%Scenario%")
= sum(CropResManSystemGrassland,
       VCropResidues_Management_MR.l(Regions_MR,CoreGrassActivities,"Average Residues (t)",
              "Crop res N for areas (tN)",CropResManSystemGrassland,ProductionSystems,
              ProductionConditions,"%Scenario%"));
```

Similarly, derive the total grass residue P available to be left on the field, per geographic unit (for details, see the code file).

1.3) Mineral fertilizers

Total mineral N and P fertilizer requirements per geographic unit are derived by either using values as specified in the scenario specifications, or by scaling the baseline total mineral fertilizer quantities by the changes in total N and P requirements, either for cropland only or for cropland PLUS temporary grassland (this can be chosen at the beginning of steering file 2, cf. section 2.3). Thus, the Baseline quantity is multiplied by the requirements in the scenario and divided by the baseline requirements.

This results in the following mineral fertilizer quantities, at this stage available for **“AllProdSyst” only if !NO! differentiation between conv/org is done**:

```
VMineralFertilizerQuantity_MR.l(Regions_MR,
       "mineral N fert (N)","AllMinFertProdTech","AllProdSyst","%Scenario%")

VMineralFertilizerQuantity_MR.l(Regions_MR,
       "mineral P fert (P2O5)","AllMinFertProdTech","AllProdSyst","%Scenario%")
```

And the mineral N fertilizer quantity is **ALSO** available for **"Convent", !IF A! differentiation between conv/org is done**:

```
VMineralFertilizerQuantity_MR.l(Regions_MR,
       "mineral N fert (N)","AllMinFertProdTech","Convent","%Scenario%")
```

For details, see the code file.

```
- 2) Relative nutrient requirements for the different crops and grasses
```

Derive various types of N and P requirements per crop in relation to total N and P requirements of all crops (or all crops PLUS temporary grassland, etc.). This parameter is then used below as a proportionality factor to allocate total available nutrients to single crops. For details, see the code file.

```
- 3) Fertilizer allocation to crops and grassland
```

The allocation of N in manure, residues and mineral fertilizers is derived by allocating the respective total quantity available per geographic region to the single crops in proportion to their relative N requirement as a share of the N requirement of all crops in this region. P from mineral fertilizers is also allocated like this. P in manure and residues is derived from the quantity of manure and residues applied due to the allocation rule for N application, because the quantity of P in manure and residues is linked to the quantity of N applied and cannot be chosen independently anymore. Thus, when applying a certain amount of N in manure and residues, this comes with a certain amount of biomass, which contains a certain amount of P. For details, see the code file.

To run these calculations, the following inputs **may/likely need to be specified** in the scenario specifications (all other values are available via the initialization (cf. section 3.2.6); but also they can be changed in the scenario specifications, if needed). These quantities can also be derived in the code, but for some scenarios, they will be provided in the scenario specifications.

- the mineral N and P fertilizer quantities:

```
VMineralFertilizerQuantity_MR.l(Regions_MR,
       "mineral N fert (N)","AllMinFertProdTech","AllProdSyst",
              "%Scenario%")

VMineralFertilizerQuantity_MR.l(Regions_MR,
       "mineral P fert (P2O5)","AllMinFertProdTech","AllProdSyst",
              "%Scenario%")
```

and also, if differentiation between org/convent is done:

```
VMineralFertilizerQuantity_MR.l(Regions_MR,
       "mineral N fert (N)","AllMinFertProdTech","Convent",
              "%Scenario%")
```

3.2.18 _V6_CoreModelEquations_FertilizerApplicationEmissions

This file contains the equations for the emissions from fertilizer application. The table of contents is:

```
$ontext;
- 1) Direct N2O emissions and N losses from fertilizer application
         1.1) Direct N2O emissions from fertilizer application
         1.2) Corresponding direct N losses from fertilizer application:
- 2) Indirect N2O emissions and N losses from fertilizer application
         2.1) Indirect N2O emissions from fertilizer application
         2.2) Corresponding indirect N losses from fertilizer application:
- 3) Adapt N deposition to manure quantities and fertilizer use
$offtext;
```

In more detail:

```
- 1) Direct N2O emissions and N losses from fertilizer application
         1.1) Direct N2O emissions from fertilizer application
```

First, some preparatory calculations on manure application are executed, to simplify the data, which is quite large for this parameter. Currently, this consists in replacing the activity dimension by “AllCrops” and “Grass” only, as the values are identical for all crop activities and for all grass activities, and it is thus not efficient to retain all these identical values for all these different activities. For details, see the code file.

Then, the emissions from fertilizer application are derived: direct N_2O emissions, emissions from volatilization and leaching, NH_3 emissions (as part of the emissions from volatilization), and corresponding N losses from direct emissions, and from volatilization and leaching. All these are provided for mineral fertilizers, manure and residues. They are all derived based on the emission factors from (IPCC 2006), emissions per ton N applied, and the amount of N applied per hectare. In the following, we list the parameters derived here, using the example of mineral fertilizer (similar for manure, residues). For further details, see the code file.

Direct N_2O emissions from fertilizer application, units: t CO_2eq / ha

```
ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "Direct N2O from mineral fert N applic (tCO2e)",ProductionSystems,
        ProductionConditions,"%Scenario%")
```

```
         1.2) Corresponding direct N losses from fertilizer application:
```

Direct N losses from fertilizer application, units: t N / ha

```
ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "Direct N loss from mineral fert N applic (tN)",ProductionSystems,
        ProductionConditions,"%Scenario%")
```

```
- 2) Indirect N2O emissions and N losses from fertilizer application
         2.1) Indirect N2O emissions from fertilizer application
```

Indirect N_2O emissions from volatilization from fertilizer application units: t CO_2eq/ha

```
ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "N2O volat from mineral fert N applic (tCO2e)",ProductionSystems,
        ProductionConditions,"%Scenario%")
```

Do NH_3 emissions as specific part of the volatilization, units: t NH_3/ha

```
ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "NH3 from mineral fert N applic (tNH3)",ProductionSystems,
        ProductionConditions,"%Scenario%")
```

Indirect N_2O emissions from leaching from fertilizer application units: t CO_2eq/ha

```
ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "N2O leach from mineral fert N applic (tCO2e)",ProductionSystems,
        ProductionConditions,"%Scenario%")
```

```
         2.2) Corresponding indirect N losses from fertilizer application:
```

N losses from volatilisation and leaching from fertilizer application. As with the emissions from manure management (section 3.2.15), this is not only the N that is lost via N_2O, but ALL N that is lost via volatilization and leaching; units: t N/ha

```
ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "N volat from mineral fert N applic (tN)",ProductionSystems,
        ProductionConditions,"%Scenario%")

ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,
        "N leach from mineral fert N applic (tN)",ProductionSystems,
        ProductionConditions,"%Scenario%")
```

To run these calculations, **no inputs need to be specified** in the scenario specifications (as all values are available via the initialization (cf. section 3.2.6); but they can be changed in the scenario specifications, if needed).

3.2.19 _V6_DeriveAggregateImpacts_PerUnit

This code calculates various totals (on a per unit (i.e. ha, head, etc.) basis), such as the sum of all GHG emissions related to cropland use (i.e. fertilizer application, deforestation, etc.), or to animals (i.e. enteric fermentation plus manure management). On this level not yet including the embodies emissions in inputs.

Currently, the following parameters are calculated; for details, see the code file:

```
        ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"Total GHG em. - crops, incl. defor/orgSoils (tCO2e)",ProductionSystems,ProductionConditions,Scenarios)
        ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"Total GHG em. - crops, no defor/orgSoils (tCO2e)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Total N-losses from animals, - manure management (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Total N-losses from animals, - man man + housing (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Total N-losses from animals, - grazing (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Total N-losses from animals, - grazing + housing (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Manure Man GHG emissions - animals (tCO2e)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"Total GHG emissions - animals (tCO2e)",ProductionSystems,ProductionConditions,Scenarios)
        ActCropsGrass_Inputs_MR(Regions_MR,Activities_MR,"N inputs - crops (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActCropsGrass_Outputs_MR(Regions_MR,Activities_MR,"N outputs - crops (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"N balance - plot level based (tN)",ProductionSystems,ProductionConditions,Scenarios)
        ActCropsGrass_OtherChar_MR(Regions_MR,Activities_MR,"NH3 from areas (tNH3)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"NH3 from animals, - manure management (tNH3)",ProductionSystems,ProductionConditions,Scenarios)
        ActAnimalsHead_OtherChar_MR(Regions_MR,Activities_MR,AnimalTypeInHerd,"NH3 from animals, - grazing + housing (tNH3)",ProductionSystems,ProductionConditions,Scenarios)
```

3.2.20 _V6_DeriveTotalImpacts

This file contains the code to derive the TOTAL impacts from the impacts per unit, i.e. per ha, per ton and per animal head or per APU values, by multiplication of the per unit impacts with the activity quantities, i.e. number of hectares, tons, animal heads, or APUs. For details, see the code file.

3.2.21 _V6_DeriveGeographicAggregations

This code sums the various total impacts over geographic groups (e.g. deriving values for “Southern Europe”, “Western Africa” or “World” by summing the corresponding values for all countries in these regions). For details, see the code file.

3.2.22 _V6_DeriveActivityGroupAggregations

This code sums the various total impacts over activity groups (e.g. deriving values for “Cereals”, “Treenuts” or “Ruminants” by summing the corresponding values for all activities in these activity groups. For details, see the code file.

3.2.23 _V6_DerivePerAPUValues

This code derives per animal production unit (APU) input, output and other values, based on the corresponding per animal head values and the herd structure. Thus, it is a weighted sum of the per animal head values with the relative share of the respective herd type animals in the APU. For details, see the code file.

3.2.24 _V6_DerivePerPrimaryProductImpacts

This file contains the code to derive per primary product values for inputs, outputs, other characteristics (focusing on environmental impacts) from the corresponding per activity values. Thus, for example, it derives per ton values from per hectare and yield values. For details, see the code file.

This and the following file are looped to derive values for various allocation and functional unit choices, both referring to mass, protein, calories and economic value.

3.2.25 _V6_DerivePerCommodityImpacts

This file contains the code to derive per commodity values for inputs, outputs, other characteristics (focusing on environmental impacts) from the corresponding per primary product values, by means of the extraction rates, etc. For details, see the code file.

This and the previous file are looped to derive values for various allocation and functional unit choices, both referring to mass, protein, calories and economic value.

3.2.26 _V6_OutputFiles_SteeringFile2

This file contains the code to produce the following gdx-files. See the code file for details on what is contained in each of those files. Currently, the following files are generated:

- GeneralModelSets_MR.gdx
- GeneralModelParameters_Inputs_MR.gdx
- GeneralModelParameters_Outputs_MR.gdx
- GeneralModelParameters_OtherChar_MR.gdx
- GeneralModelParameters_Various_MR.gdx
- GeneralModelParameters_Auxiliary_MR.gdx
- GeneralModelVariables_ActivityQuantities_MR.gdx
- GeneralModelVariables_Inputs_MR.gdx
- GeneralModelVariables_Outputs_MR.gdx
- GeneralModelVariables_OtherChar_MR.gdx
- GeneralModelVariables_Various_MR.gdx
- GeneralModelVariables_Trade_MR.gdx
- GeneralModelVariables_CommodityTree_MR.gdx
- GeneralModelVariables_Auxiliary_MR.gdx

3.2.27 _V6_ResultsFiles

This file contains all the code to produce some specifically designed results files with a selection of results of interest only, for further use, e.g. to produce graphs, etc.; for details, see the code file.

To only have the scenarios of interest included in the results, it may be necessary to adapt the corresponding set-definition at the beginning of the part of the file where the results parameters Results(…, Results2(…, etc. are assigned. See as an example the following code:

```
*choose a subset of scenarios to display to avoid values to be displayed that are not of interest (e.g. from other scenarios
that are read in the model but not used here, etc.)
SET ScenariosResults(Scenarios)
/
*Baseline, BaselineDerived, FOFA_BAU_2012, FOFA_BAU_2050, FOFA_BAU_2050_Test, LfP_2050, AE_exports_2050, LfS_2050,
AE_food_2050
Baseline, BaselineDerived, zhaw_NoFeedImports
/;
```

3.2.28 __SOLmV5_CoreModelEquations_SomeSpecialOutputForNFP69.gms

This file is currently not in use, it has to be transferred to SOLmV6 yet. It has been used to produce some additional specific output for a particular project. At this place in the steering file 2, other such files could be added, for some final calculations for additional specific output, as needed for single projects, etc.

3.3 Output files and graphics

SOLm provides a number of output files containing all parameters and variables used in the model (cf. section 3.2.26). It also provides specific results files, containing a restricted amount of parameter and variable values only, designed for efficient further processing, e.g. analysis of few core results, or as a basis for tables and figures for specific results (cf. section 3.2.27). The results can also be read out to excel-files, where graphics can be readily produced. If such figures are predefined in excel, GAMS can write new data to the excel-files, such as to readily generate the pre-defined graphics with this new data.

3.4 Scenario specification

The general structure of the scenario assumption code file and the variables and parameters that have to or likely are assigned by scenario specifications are described and listed in section 3.2.8.

Here, we display for illustration the assumptions for the scenario "BaselineDerived" which derives all baseline values from setting the cropping area numbers only and for the scenario "Baseline_100Organic" which switches the baseline situation to 100% organic production, based on a number of additional assumptions besides areas. These are relatively simple sets of scenario specifications, and many more assumptions on parameters and variables can be added, on any aspects that may be changed in a scenario – for details see the code file "`_V6_ScenarioSpecifications.gms`":

Further examples will be added later..

Baseline derived:
```
*1) BaselineDerived
$label AssumptionsBaselineDerived

$setglobal UseProdSyst_ConvOrg "YES"

*assign areas (no organic areas)
VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,ProductionSystems,
       ProductionConditions,"%Scenario%")
= VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,ProductionSystems,
       ProductionConditions,"Baseline");

VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"Convent",
       ProductionConditions,"%Scenario%")
= VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"AllProdSyst",
       ProductionConditions,"Baseline");

$goto EndOfScenarioSpecifications
```

Baseline 100% organic:

```
*8) Baseline_100Organic
$label AssumptionsBaseline_100Organic

$setglobal UseProdSyst_ConvOrg "YES"

*assign areas as already assigned, but 0% convent, 100% organic:
VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"Convent",
```

```
        ProductionConditions,"Baseline_100Organic")
          = 0;

VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"Organic",
        ProductionConditions,"Baseline_100Organic")
  = VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"AllProdSyst",
        ProductionConditions,"Baseline");

VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"AllProdSyst",
        ProductionConditions,"Baseline_100Organic")
  = VActCropsGrass_QuantityActUnits_MR.l(Regions,Activities,"AllProdSyst",
        ProductionConditions,"Baseline");


*adjust legume area shares in case they are less than 20% in total:
VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Legumes (Nfixing)",
        ProductionSystems,ProductionConditions,"Baseline")
  =sum(Legumes_NFixing,VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Legumes_NFixing,
        ProductionSystems,ProductionConditions,"Baseline"));

VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops without Legumes (Nfixing)",
        ProductionSystems,ProductionConditions,"Baseline")
  =sum(Crops_NoNFixingLegumes,VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,
        Crops_NoNFixingLegumes,ProductionSystems,ProductionConditions,"Baseline"));

VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops",ProductionSystems,
        ProductionConditions,"Baseline")
  =sum(Crops_MR,VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Crops_MR,
        ProductionSystems,ProductionConditions,"Baseline"));

ActCropsGrass_OtherChar_MR(Regions_MR,"All Crops","Share legumes in cropland (share ha)",
        ProductionSystems,ProductionConditions,"Baseline")
     $VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops",ProductionSystems,
= VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Legumes (Nfixing)",
        ProductionSystems,ProductionConditions,"Baseline")
     /VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops",ProductionSystems,
                ProductionConditions,"Baseline");

VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Legumes_NFixing,"Organic",
        ProductionConditions,"Baseline_100Organic")
     $(VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Legumes (Nfixing)",
        "AllProdSyst",ProductionConditions,"Baseline")
       AND (ActCropsGrass_OtherChar_MR(Regions_MR,"All Crops",
        "Share legumes in cropland (share ha)","AllProdSyst",
                ProductionConditions,"Baseline") < 0.2))
  = 0.2*VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops","AllProdSyst",
        ProductionConditions,"Baseline")
                *VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Legumes_NFixing,"AllProdSyst",
                        ProductionConditions,"Baseline")
                   /VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Legumes (Nfixing)",
                        "AllProdSyst",ProductionConditions,"Baseline");

VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Crops_NoNFixingLegumes,"Organic",
        ProductionConditions,"Baseline_100Organic")
     $(VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops without Legumes (Nfixing)",
        "AllProdSyst",ProductionConditions,"Baseline")
       AND (ActCropsGrass_OtherChar_MR(Regions_MR,"All Crops",
        "Share legumes in cropland (share ha)","AllProdSyst",
                ProductionConditions,"Baseline") < 0.2))
  = 0.8*VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,"All Crops","AllProdSyst",
        ProductionConditions,"Baseline")
                *VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,Crops_NoNFixingLegumes,
                        "AllProdSyst",ProductionConditions,"Baseline")
                   /VActCropsGrass_QuantityActUnits_MR.l(Regions_MR,
                        "All Crops without Legumes (Nfixing)",
                        "AllProdSyst",ProductionConditions,"Baseline");


*assign mineral N fertilizer values:
```

```
VMineralFertilizerQuantity_MR.l(Regions,"mineral N fert (N)","AllMinFertProdTech",
       "AllProdSyst","Baseline_100Organic")
  = VActCropsGrass_Inputs_MR.l(Regions,"All crops and grass","N from all fertilizers (tN)",
       "AllProdSyst","AllProdCond","Baseline")*0;
VMineralFertilizerQuantity_MR.l(Regions,"mineral N fert (N)","AllMinFertProdTech",
       "Convent","Baseline_100Organic")
  = VActCropsGrass_Inputs_MR.l(Regions,"All crops and grass","N from all fertilizers (tN)",
       "AllProdSyst","AllProdCond","Baseline")*0;

$goto EndOfScenarioSpecifications
```

3.5 Adding new data

New data is added in several steps towards the end of steering file 1 (cf. beginning of section 3.1), captured in specific files for organizing this (some examples are given further down in this section).

First, by adding better values of already existing ones, e.g. better country specific values on animal numbers and cropping areas than reported in FAOSTAT. This is done by replacing the old values with the better ones by assigning the latter to the corresponding parameters or variables.

Second, by adding new inputs, output, other characteristics, etc. to the variables and parameters, defining these new elements in the corresponding sets and then assigning the corresponding values.

Third, by defining new regions and activities and herd-structures, if needed (e.g. sub-regions of countries or sub-activities of existing activities or new activities, or new herd animal types) and then assigning data to these new regions and activities and herd animal types. The default for parameters values for new regions and activities is to assign existing data from the baseline or reference scenario values, as it fits (e.g. country values for parameters on sub-country levels). The default for total values for new regions and activities (variables) is to allocate them proportionally to some other variable (e.g. imports to sub-regions could be allocated proportionally to population). Where specific and better data for these total values is available, this is directly assigned, replacing these default values.

These new indicators etc. can also be derived from e.g. gridded data, process models on plot or farm level, etc. Thus, SOLm can link to such more refined models if their output is aggregated adequately.

Data for new herd structures is assigned similarly, by assigning parameter values from existing herd structure information that fits best (if no better data is directly provided from some new data set to be used), and by assigning variable values proportionally (similar to sub-regions, etc. as described above, in case the new herd structure refines the old one), or by assigning values from the best fitting category – of, if specific better data is available, by directly assigning this.

The following sections present some examples of files for reading new data. For details, see each of the respective code files.

3.5.1 _V6_ReadAdditionalData_SwitzerlandAustria

This code reads new more detailed data for Switzerland and Austria, as used in a completed project on the alpine region.

3.5.2 _V6_ReadAdditionalData_NUTS2_EU

This code reads new more detailed data for NUTS2 data for the EU as used in the H2020 project UNISECO.

3.5.3 _V6_ReadAdditionalData_AnimalNumbers

This code reads new more detailed data on animal numbers incl. herd structure for EU countries (from Eurostat).

3.5.4 _V6_ReadAdditionalData_GHGInventories

This code reads new more detailed data to calculate national greenhouse gas inventories. Currently, this is data from the Swiss and Austrian inventory, which is then also used to do consistency checks as SOLm needs to replicate the emission numbers reported in the inventories.

3.5.5 _V6_ReadAdditionalData_Prices

This code reads new more detailed data for commodity prices. Currently, this is one set of prices from a Swiss retailer, to have a somewhat better database than the FAOSTAT producer prices for Switzerland. This is used in one specific project only (NFP69, Sustainable and healthy diets).

3.5.6 _V6_ReadAdditionalData_FeedData

This code reads new more detailed data on feed imports and availability. Currently, this covers feed imports to Switzerland only, as used in some calculations in a small project on sustainable agriculture in Switzerland by the zhaw.

3.5.7 _V6_ReadData_FAOSTAT_FOFA2050

This code reads the data for the food system projections to 2050 as presented in (FAO 2018).

3.5.8 _V6_ReadData_VariousSources_BioenergySR15

This code reads the data for the energy and food system projections to 2050 and 2100 as presented in (IPCC 2018), containing scenarios that massively rely on bioenergy with carbon capture and storage (BECCS).

3.6 Herd structures

This section describes the herd structures used in SOLm (cf. section 3.1.43). It is based on the supplementary information of (Schader, Muller et al. 2015). For the EU, these values are replaced by the better values directly taken from Eurostat, cf. section 3.5.3.

Herd structures describe the composition of an animal herd in terms of different sub-classes defined according to age, sex and production purpose of the animal (Table 1). Herd structures were calculated for pigs and cattle based on assumptions relating to fertility rates, age of first calving, slaughtering rates and losses due to diseases and accidents (Table 2), calibrated using FAOSTAT data for producing animals, living animals (stocks), imports and exports. Herd structures were calculated for each country with a separate optimization model using a cross entropy estimator (Golan, Judge et al. 1996). These models predict the most likely average herd structure in a country, based on information that is available. Support points 1, 2 and 3 were defined based on expert opinions. While support point 2 describes a central value, support points 1 and 3 refer to the upper and lower bounds of a country-specific parameter.

$$H(p,q) = -\sum_{X} p(x) * \log q(x)$$

Where:
$H(p,q)$ is the entropy
$P(x)$ is the true distribution
$q(x)$ is the given probability distribution

This model has been run with baseline data averaging the years 2005-2009.

Table 1. Overview of livestock types defined in cattle, pig and chicken herd structure models

Herd	Sub-class	Description
Cattle	A_Dcow	Number of dairy cows
Cattle	A_Dsire	Number of dairy sires
Cattle	A_DFemaleCalf1	Number of female dairy calves aged 1 year
Cattle	A_DFemaleCalf2	Number of female dairy calves aged 2 years
Cattle	A_DFemaleCalf3	Number of female dairy calves aged 3 years
Cattle	A_DMaleCalf1	Number of male dairy calves aged 1 year
Cattle	A_DMaleCalf2	Number of male dairy calves aged 2 years
Cattle	B_Bcow	Number of beef cows
Cattle	B_Bsire	Number of beef sires
Cattle	B_BFemaleCalf1	Number of female beef calves aged 1 year
Cattle	B_BFemaleCalf2	Number of female beef calves aged 2 years
Cattle	B_BMaleCalf1	Number of male beef calves aged 1 year
Cattle	B_BMaleCalf2	Number of male beef calves aged 2 years
Pigs	A_Sows	Number of sows
Pigs	A_Boars	Number of boars
Pigs	A_Sucklers	Number of sucklers
Pigs	A_Weaners	Number of weaners
Pigs	A_Fatteners	Number of fatteners

Table 2. Overview of external variables of the cattle, pig and chicken herd structure models

Herd	Variable	Support point 1	Support point 2	Support point 3
Cattle	Share of calve losses in year 1	0.01	0.10	0.50
Cattle	Share of calve losses in year 2	0.01	0.10	0.50
Cattle	Share of calve losses in year 3	0.01	0.10	0.50
Cattle	Fertility rate of dairy cows	0.50	0.90	1.00

Herd	Variable	Support point 1	Support point 2	Support point 3
Cattle	Fertility rates of beef cows	0.50	0.80	1.00
Cattle	Calving rates	0.55	0.95	1.00
Cattle	Share of slaughtered male dairy calves aged 1 year	0.00	0.05	0.10
Cattle	Share of slaughtered female dairy calves aged 1 year	0.00	0.01	0.10
Cattle	Share of slaughtered female dairy calves aged 2 years	0.00	0.05	0.10
Cattle	Share of slaughtered male beef calves aged 1 year	0.00	0.05	0.10
Cattle	Share of slaughtered female beef calves aged 1 year	0.00	0.05	0.10
Cattle	Share of sires	0.0095	0.01	0.0105
Cattle	Dairy cow replacement rate	0.10	0.30	0.50
Cattle	Beef cow replacement rate	0.10	0.30	0.50
Cattle	Dairy sire replacement rate	0.10	0.30	0.50
Cattle	Beef sire replacement rate	0.10	0.30	0.50
Cattle	Age at first calving (years)	2.00	2.50	4.00
Pigs	Share of suckler losses	0.01	0.12	0.40
Pigs	Share of weaner losses	0.01	0.05	0.20
Pigs	Share of fattener losses	0.01	0.03	0.10
Pigs	Suckling period (days)	7.00	28.00	45.00
Pigs	Weaning period (days)	20.00	35.00	60.00
Pigs	Fattening period (days)	80.00	120.00	250.00
Pigs	Litters per year	1.00	1.50	2.50
Pigs	Litter size	8.00	13.00	15.00
Pigs	Share of boars	0.01	0.08	0.10
Pigs	Culling rate of sows	0.20	0.40	0.50
Pigs	Culling rate of boars	0.25	0.40	0.50
Pigs	Age at first parturition (months)	12.00	15.00	18.00

All herd structures are based on a virtual starting population of each age and sex category (marked in Figure 4, Figure 5 and Figure 6 with the abbreviation ST). After one year the starting population is modified by: a) subtracting young animal losses (LO), animal slaughtering (SL), young animals entering the adult population (XC), and exports (EXP); and b) adding replacement animals (REP) and imports (IMP), to give an end of year population. The final number of animals in each category refers to the arithmetic mean of the start (ST) and end population numbers (EN). The end population of each cohort serves as the starting population for the next year (e.g. the end population of one year old animals in the first year serves as the starting population of two year old animals in the second year). For pigs of different age and sex categories, we calibrated the herds to one year. Average numbers (AV) are multiplied by the number of life cycles per year. For cattle, we defined a dairy and a beef herd structure. Each herd structure is linked through several conditions (purple boxes) which state that the total number slaughtered, living, imported and exported must equal the observed quantities in FAOSTAT.

Figure 4, Figure 5 and Figure 6 describe the herd structures and dynamics for dairy cattle, beef cattle and pigs. All herd structure figures should be read from left to right for each age and sex category. The dairy and beef herd structures are similar, except for the absence of third year female beef calves. Some support points also differ between dairy and beef cattle (Table 2).

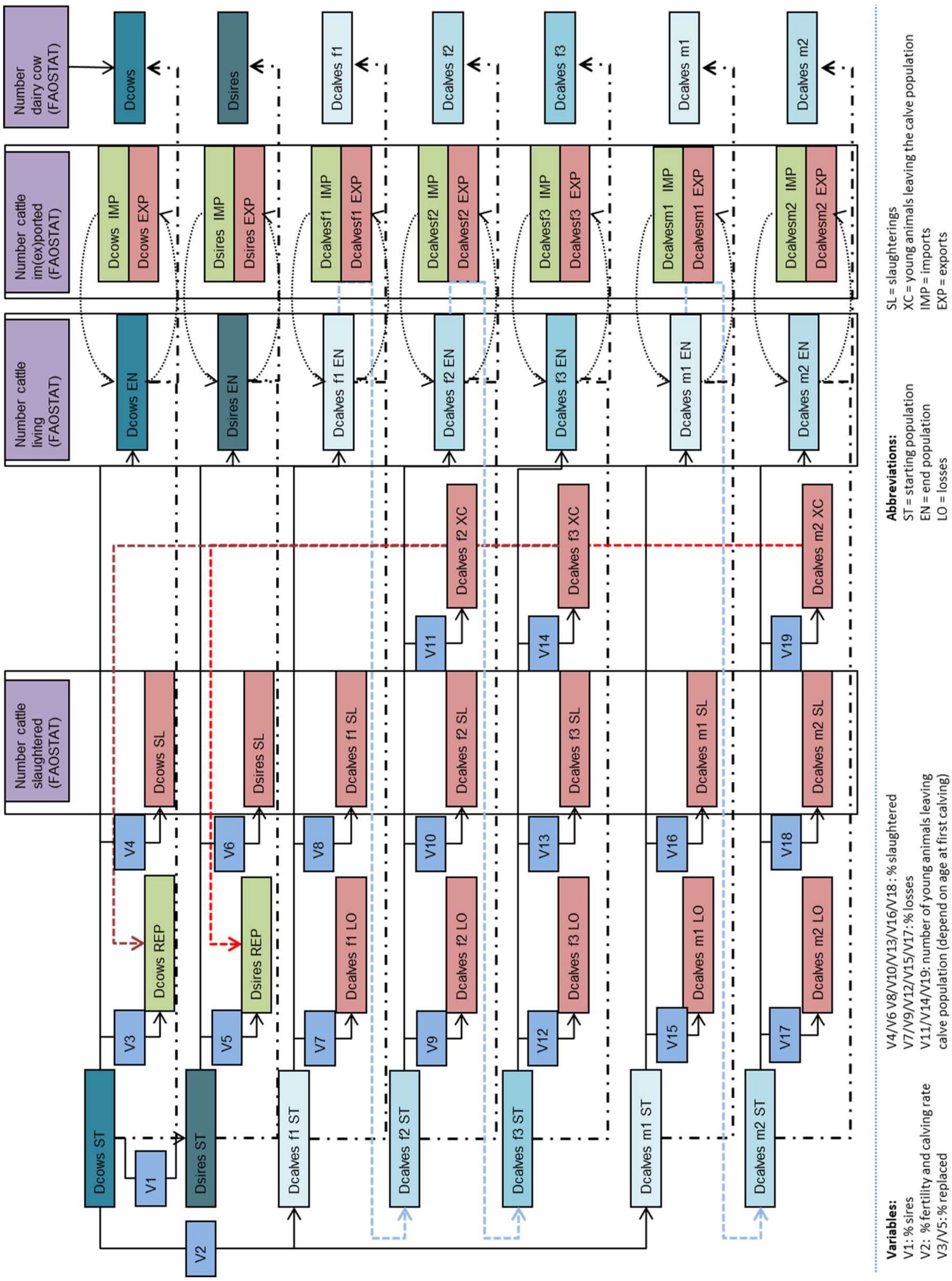

Figure 4. Illustration of dairy cow herd structure

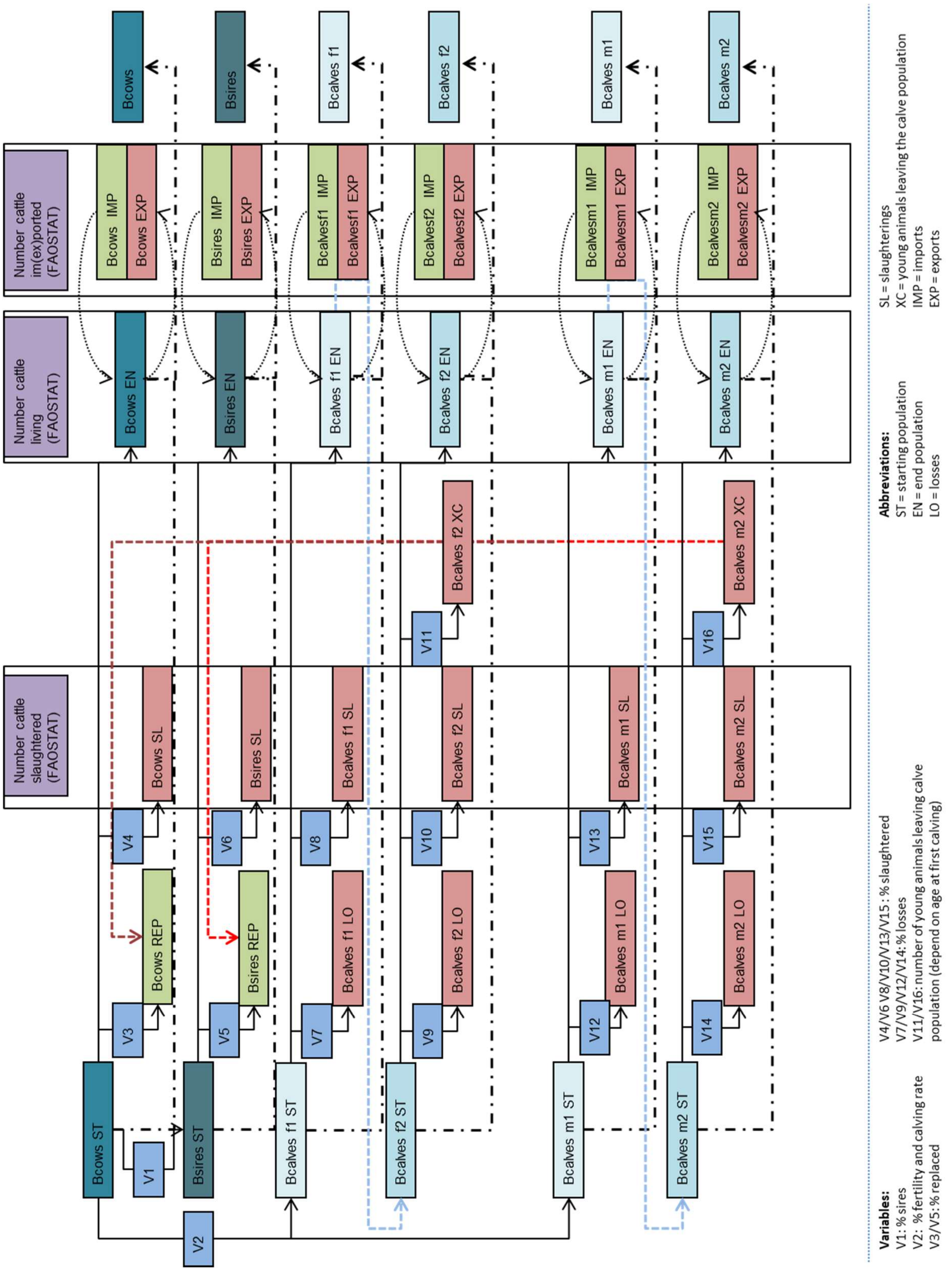

Figure 5. Illustration of beef cow herd structure

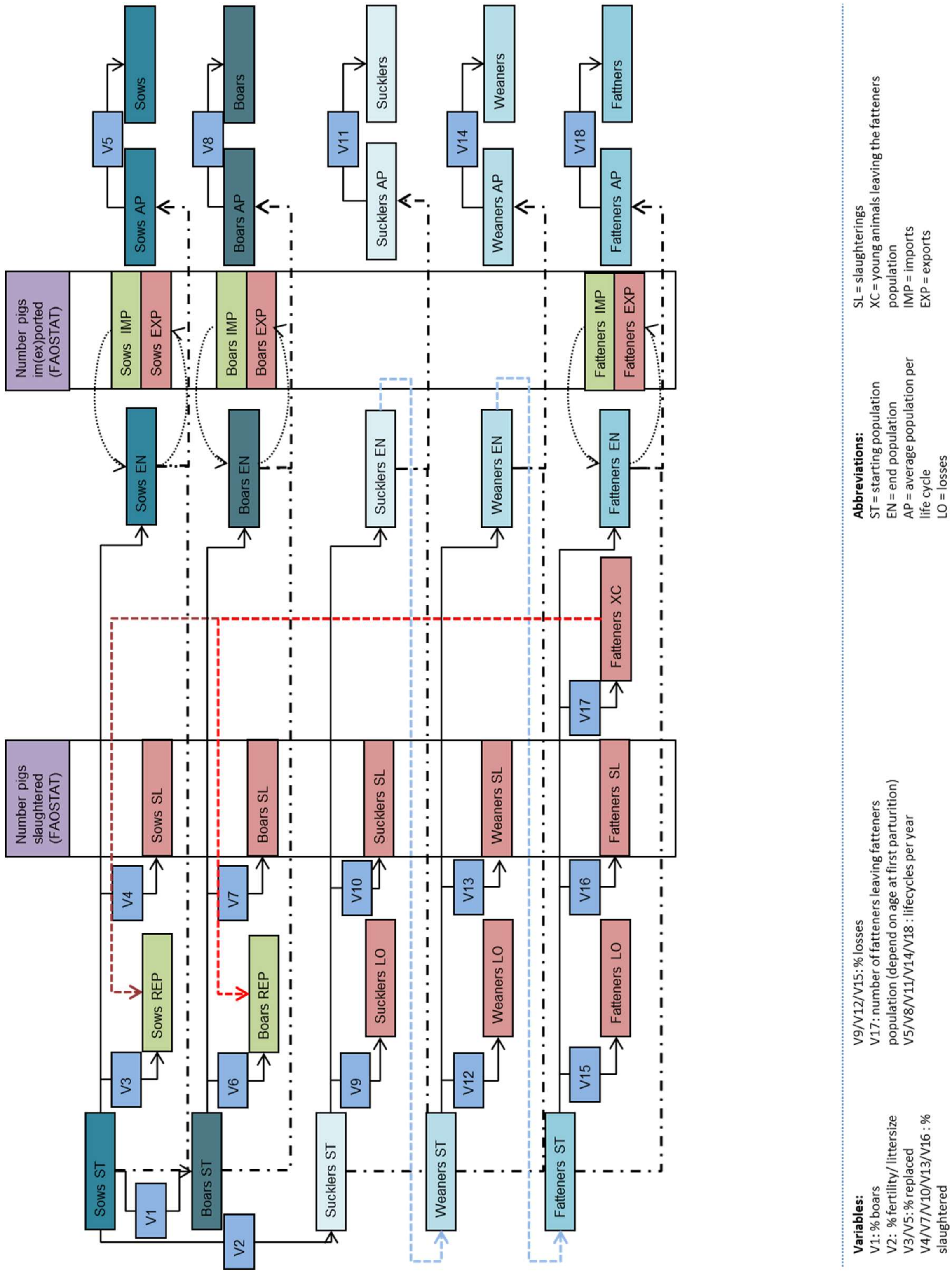

Figure 6. Illustration of pig herd structure

3.7 Feeding rations and feed supply for animals

See also section 3.2.13. Currently, SOLm contains assumptions on feeding rations on the level of single commodities (on the basis of the share of each commodity in total DM intake per head): FeedingRationsHeads_OtherChar_MR(Regions_MR,Activities_MR, AnimalTypeInHerd,Commodities_Feed,"Quantity share in DM (share)", ProductionSystems,ProductionConditions,"%Scenario%").
These values are derived from information on feeding rations based on four feed groups, namely concentrates, forage crops, grass and residues, that are then scaled down to single commodities constituting these groups. These feed groups are built from various commodities as defined in the corresponding set in section 3.1.15.14 (Match_FeedCommoditiesToFeedCommodGroups(Commodities,Commodities_2) and Match_FeedCommoditiesToFeedCommodGroups_MainByprodConc(Commodities,Commodities_2) differentiating between main products and byproducts for concentrate feed (second set) or not (first set)). Defaults for the detailed feeding rations, i.e. for feeding rations on single commodity level are assigned in section 4.4. of the code module `_V6_BaselineValues_ForModelRuns` (cf. section 3.2.5).
The feeding rations based on single commodities allow to have differentiated feed composition per animal type, e.g. different concentrate feed composition for pigs and dairy cattle, etc.
From the feeding rations and the feed requirements, it is then possible to derive the detailed commodity supply per animal head and the related impacts of these commodity quantities and thus of the animal feed.
The feed commodities are also grouped in more refined commodity groups, as detailed in Match_FeedCommodToFeedCommodGroups_DetailedFeedRatio, encompassing groups such as grass, cereals, pulses, forage, etc.
Currently, animal numbers are derived based on total grass and cereal supply, where the aggregates of those are currently not differentiated per animals. To derive how many animals can be fed, thus, goats and cattle get the same mix of available grasslands, and pigs and chickens the same mix of cereals. When it comes to impacts of this feed, though, the commodity-level-differentiation is implemented, though, as the feeding rations are provided on this level.

3.8 Crop rotations

Currently, crop rotations are dealt with by assuming fixed area shares of the different crop rotation elements. Thus, instead of cropping a number of different crops one season after the other on the same plot, it is captured by cropping the corresponding shares of the area for each crop (e.g. a rotation with five different crops on one hectare in five years is captured by allocating a fifth of a hectare to each of these crops).

Thus, there is no specific nutrient dynamics between different elements of the crop rotations captured, such as from soybeans in one year to maize in the subsequent year. Residues left on the field are currently treated as other residues (e.g. in compost) regarding application to soils (management losses are however zero for residues left on the field, and emissions from those are accounted for under application losses only; cf. section 3.2.17).

Another approach to model crop rotations in SOLmV6 is to define new activities that include all elements (e.g. “Soy-maize-rotation”) and report 50% of the annual yield only – but for each crop. The specific nutrient dynamics can then be captured by reducing the nutrient requirements from maize and the residue left on the field output from soy, for example.

3.9 Fish and Seafood

In SOLmV6, there is no default data on fish and seafood production from FAOSTAT, while FAOSTAT provides default values for commodity quantities. Fish and seafood production is thus captured in specific activities with corresponding input, output and other characteristics parameters and quantity variables and the default data for this is read in the code file "`_V6_ReadData_VariousSources_FishAndSeafoodData.gms`" (section 3.1.59). In this file, the commodity quantities from FAOSTAT are then also adjusted to be consistent with the production values read in. Part of the domestically available quantities for feed are allocated to fish and seafood and then not available for livestock anymore. This is organized in code file `_V6_CoreModelEquations_NutrReqAndFeedSupply_DetailedFeedingRations.gms`" (section 3.2.13).

3.10 Bioenergy

Currently, bioenergy is captured via two activities, namely miscanthus and residues from forestry. For the former, input needs regarding nitrogen demand are specified, and for both it is derived how much nitrogen may be recycled to the fields after the various processing steps for energy generation. The data is taken from (IPCC 2018), both for the baseline (here the year 2010) and several scenarios for 2050. For details, see section 3.1.70.

3.11 Fertilizer Application

SOLm calculates the fertilizer application levels per ha based on the nutrient requirements of the plants. These nutrient requirements are either provided by baseline or reference scenario data to be read explicitly (not included in the default data), or by the nutrients in the total outputs from the activities (i.e. in the sum of the main products and residues, cf. section 3.2.17, and in detail, section 3.2.12).

The nutrients available per region are then applied proportionally to these requirements. This applies for nitrogen in manure, residues and mineral fertilizers and for phosphorus in mineral fertilizers. Phosphorus in manure and residues is calculated from the quantities of manure and residues applied, which have been derived based on the nitrogen application in these fertilizers (i.e. choosing the quantity of manure and residues based on nitrogen determines how much phosphorus is then applied, as the phosphorus quantity applied is not independent of the nitrogen quantity, both being related to the quantity of manure and residues via their contents.

The nutrients applied are thus dependent on the yield levels. Currently, yields do however not directly react to nutrient inputs. Furthermore, the nutrient inputs derived like this may not reflect values reported in the literature. This is due to these latter often being determined in optimal systems without massive over-fertilization, for example fertilization rates suggested by advisory services' institutions, while the former account for the overall nutrient availability in a region, which can be way beyond what would be required from the crops, but which is in reality nevertheless applied to the fields (e.g. nutrients in manure in regions of massive concentrate feed imports resulting in oversupply of nutrients in manure).

Nutrient supply levels thus result in an indicator of nutrient oversupply (in case the total nutrient balance of nutrient inputs minus nutrient outputs is positive) or undersupply (in the opposite case). In a situation of oversupply, the nutrients not used are largely lost to the environment. In the case of undersupply, yield levels may be kept due to some non-reported nutrient flows, e.g. from the soil pool. But in general, undersupply rather indicates that the production as modelled in this case is not possible and that yields should be lower or additional fields need to be cropped with legumes to provide sufficient nutrients.

Currently, this is taken up by reporting the nutrient undersupply as some risk level for yield drops or the yields are adjusted downwards or the legume areas are increased in additional scenario runs, to achieve a more balanced situation where nutrient supply and demand match better. This is then implemented by corresponding scenario specifications: The dependence of yields on nitrogen supply can in a first approximation be modelled as a linear relation between nitrogen inputs and yields (e.g. (Godard, Roger-Estrade et al. 2008)). This however may tend to underestimate yields. At lower supply levels, changes in yields in reaction to changes in nitrogen supply tend to be larger than at higher levels and the response curve is concave rather than linear. It thus can be captured in more detail by crop specific response functions of yields to nitrogen supply taken from the literature.

It is also important to emphasize that the "equilibrium view" adopted in SOLm, as described in section 1.3 also influences how fertilization application and nutrient availability is to be understood. Nutrient balances, etc. are calculated on the basis of the annual input and output values, but it is implicitly assumed that part of the nutrients applied are stored in the soil for one, two or more years, but that corresponding nutrient quantities are also released from the soil in the given year, one, two or more years after their earlier applications. This could be refined by adding data on nutrient storage in and release from the soils, but this is not part of SOLmV6 yet.

3.12 Trade-flow reorganization

FAOSTAT/TRADESTAT reports commodity trade between countries in considerable detail. For assessing sustainability impacts of the commodities traded, it is important to know the country of origin, which is often not directly intelligible from TRADESTAT, as trade via several trade partners may occur. SOLm implements the method suggested in (Kastner, Kastner et al. 2011) to trace commodity trade back to the countries of origin. The principle to do this is to assume that exports from country A of a commodity X originate proportionally to the production- and import-shares of the domestically available quantity (DAQ) of commodity X in country A from domestic sources or imports. Thus, the share in DAQ of production of X in country A and of the imports of X from other countries to country A, are the shares to be applied to the exports of commodity X from country A.

Details to be added and the respective code-module to be fully linked to SOLm

3.13 Linking activities and commodities

Activities and commodities are linked via the commodity trees and extraction rates. Default values for those are taken from (FAO 2000, FAO 2001). This code is organized in seven specific sets relating different types of commodities and activities. The seven sets are also listed and described shortly in section 3.2.1.

3.14 Code details for consistency checks

Here, code details for automated consistency checks will be added as soon as such are available. Currently, the consistency checks are done manually, cf. section 2.6.

To be done

3.15 Deforestation data

The following is an updated version of the description of deforestation in (Schader, Muller et al. 2015). Because agricultural land is scarce and natural grasslands are generally not well suited for cultivation (water or temperature limited), increasing the amount of land needed for agricultural production increases pressure on grasslands and forests (Smith, Gregory et al. 2010). Conversion of grassland to cropland may also indirectly lead to increased deforestation, owing to displacement effects that result in the conversion of forests to meadows and pastures (Andrade de Sá, Palmer et al. 2013, Meyfroidt, Lambin et al. 2013). With limited data available, we have assumed that additional cropland generally increases pressure on forests and may lead to increased deforestation. Following (Kissinger, Herold et al. 2012), we have attributed 80% of deforestation to agriculture.

The deforestation potential of agricultural land expansion is estimated as follows: deforestation values are calculated using the total agricultural area as a proxy for the pressure of agriculture on forests; deforestation rates are then calculated by multiplying the total agricultural land area per region in each scenario by the ratio of deforestation areas per region from FAOSTAT (Tubiello, Salvatore et al. 2013) over total agricultural land area per region in the base years, scaled by the factor 0.8. In the same way, emissions from deforestation as available in FAOSTAT (Tubiello, Salvatore et al. 2013) are linked to agricultural land. The indicators for deforestation were applied only in the cases of positive deforestation rates. Deforestation was set to zero in countries where total forest area increased.

Thus, the deforestation values and related emissions currently included in SOLmV6 serve as a gross pressure indicator, to assess how big pressure of increasing land use on deforestation in a certain region may be and which potential effect this may have on GHG emissions.

3.16 Utilization of organic soils

Agricultural utilization of organic soils leads to huge emissions (Leifeld and Menichetti 2018). These are accounted for in SOLm as follows. The potential of agricultural land use and expansion to utilize organic soils and result in corresponding emissions is estimated as follows: organic soil utilization values are calculated using the total agricultural area as a proxy for the pressure of agriculture on organic soils; organic soil utilization rates are then calculated by multiplying the total agricultural land area per region in each scenario by the ratio of organic soil areas under agriculture per region from FAOSTAT (Tubiello, Salvatore et al. 2013) over total agricultural land area per region in the base years. In the same way, emissions from utilization of organic soils (CO_2, N_2O) as available in FAOSTAT (Tubiello, Salvatore et al. 2013) are linked to agricultural land.

Thus, the utilization of organic soils and related emissions currently included in SOLmV6 serve as a gross pressure indicator, to assess how big pressure of increasing land use on utilization of organic soils in a certain region may be and which potential effect this may have on GHG emissions.

3.17 Irrigation water use data

This section describes the irrigation water use data from AQUASTAT (AQUASTAT 2019) and the Water Footprint Network (Water Footprint Network 2019) and others (Pfister et al.), cf. 3.1.39. This data contains irrigation water use values in m^3/ha for various crops and countries, shares of irrigated areas per crop and country, as well as water scarcity indicators as described in in more detail.

Yet to be completed

3.18 Animal welfare data

This section describes the animal welfare data used in SOLm, cf. 3.1.40. This covers indicators for general health levels in different livestock production systems, such as related to intensity and yields (e.g. an index for mastitis incidence in different dairy production systems, or for parasites infestation in pasture-based systems). Other indices capture the possibilities for showing natural behaviours, such as an index for roaming space, livestock density or also having horns or not. In total, these indicates provide a pressure/risk indicator for decent welfare and living conditions for livestock, i.e. capturing whether a change to certain production systems may rather increase risk, that animal welfare and health deteriorate or not.

Yet to be completed

3.19 Pesticides data

This section describes the qualitative aggregated pesticide use indicator used in SOLm (cf. section 3.1.41). It is based on the supplementary information of (Schader, Muller et al. 2015). It is built based on i) pesticide use intensity per crop and farming system, ii) pesticide legislation in a country, and iii) access to pesticides by farmers in a country, as displayed in *Table **3***, *Table **4*** and *Table **5***. These values are based on expert judgments from 2012:

Table 3. Pesticide model classifications

Rating	Pesticide level per crop (PUI)
0	No harmful pesticides* used
1	Low level of pesticide application
2	Medium level of pesticide application
3	High level of pesticide application / harmful pesticides used*
	* WHO classification

Rating	Pesticide legislation per country (PL)
0	All chem.-synthetic pesticides (WHO-classes 1-2) banned
1	Rigid pesticide legislation and control excludes harmful pesticides*
2	Average pesticide legislation and control
3	Legislation does not preclude the use of harmful pesticides*
	*WHO classification

Rating	Access to pesticides per country (AP)
0	Farmers have no access to chem.-synthetic (WHO-classes 1-2) pesticides
1	Only few farmers have access to chem.-synthetic pesticides (max. 10% of the cultivated land is treated with pesticides)
2	Some farmers have access to chem.-synthetic pesticides (10-50% of the land that deserves treatment is treated)
3	Many farmers have access to chem.-synthetic pesticides (min. 50% of the land that deserves treatment is treated)

Table 4. Country-specific ratings of pesticide legislation (PL) and the accessibility of pesticides to farmers (AP)

Country	PL	AP
Afghanistan	3	1
Albania	2.5	2.5
Algeria	2	2
American Samoa	2	3
Andorra	2	3
Angola	3	1
Anguilla	2	3
Antigua and Barbuda	2	3
Argentina	2	2.5
Armenia	3	3
Aruba	2	3
Australia	1	3
Austria	1	3
Azerbaijan	3	2
Bahamas	2	3
Bahrain	2	2.5
Bangladesh	3	1
Barbados	2	3
Belarus	2.5	3
Belgium	1	3
Belize	2	3
Benin	2.5	1
Bermuda	2	3
Bhutan	3	1.5
Boli*via* (Plurinational State of)	3	2
Bosnia and Herzegovina	2	3
Botswana	2	1.5
Brazil	1.5	2.5
British Virgin Islands	2	3
Brunei Darussalam	2	2
Bulgaria	2.5	3
Burkina Faso	2.5	1
Burundi	2	2
Cambodia	3	1
Cameroon	3	2
Canada	1	3
Cape Verde	3	1.5
Cayman Islands	2	3
Central African Republic	3	1
Chad	3	1
Channel Islands	1	3
Chile	2	3
China	3	2
Colombia	2	2.5
Comoros	2	2
Congo	3	1
Cook Islands	1.5	3
Costa Rica	2	3
Côte d'Ivoire	3	1
Croatia	1.5	3
Cuba	2.5	2
Cyprus	2	3
Czech Republic	2	3
Democratic People's Republic of Korea	3	1.5
Democratic Republic of the Congo	3	1
Denmark	1	3
Djibouti	3	2
Dominica	2	2.5
Dominican Republic	2	2.5
Ecuador	2.5	2.5
Egypt	2	3
El Salvador	2	3
Equatorial Guinea	3	2
Eritrea	2.5	1
Estonia	1.5	3
Ethiopia	2	1.5
Falkland Islands (Malvinas)	2	3
Faroe Islands	1	3
Fiji	2	3
Finland	1	3
France	1	3
French Guiana	2	2
French Polynesia	1.5	3
Gabon	3	1.5
Gambia	2.5	1
Georgia	3	2
Germany	1	3
Ghana	2	1.5
Gibraltar	2	3
Greece	2	3
Greenland	1	2.5
Grenada	2	3
Guadeloupe	2	2.5
Guam	2	3
Guatemala	2	3
Guinea	2.5	1
Guinea-Bissau	3	1
Guyana	2.5	2
Haiti	3	2
Honduras	2	3
Hungary	2	3
Iceland	1	3
India	3	1.5
Indonesia	3	1
Iran (Islamic Republic of)	3	2
Iraq	2	2
Ireland	1	3
Isle of Man	1	3
Israel	2	3
Italy	1.5	3
Jamaica	2	2.5
Japan	1	3
Jordan	2	3
Kazakhstan	3	2
Kenya	2	2
Kiribati	2	3
Kuwait	2	3
Kyrgyzstan	3	2
Lao People's Democratic Republic	3	1
Lat*via*	1.5	3
Lebanon	2	2
Lesotho	2	2
Liberia	3	1
Libya	2	2
Liechtenstein	2	3
Lithuania	1.5	3
Luxembourg	1	3
Madagascar	2	2
Malawi	2	2
Malaysia	2	1.5
Maldives	2	2
Mali	2.5	1
Malta	1.5	3
Marshall Islands		
Martinique	2	3
Mauritania	2.5	1
Mauritius	2	2
Mayotte	3	1
Mexico	2	2.5
Micronesia (Federated States of)	2	3
Mongolia	3	1
Montenegro	2	3
Montserrat	2	3
Morocco	1	2
Mozambique	2	2
Myanmar	3	1
Namibia	2	2
Nauru	2	3
Nepal	3	1
Netherlands	1	3
Netherlands Antilles	2	3
New Caledonia	1.5	3
New Zealand	1	3
Nicaragua	2	2.5
Niger	2.5	1
Nigeria	3	1
Niue	2	2.5
Norfolk Island		
Northern Mariana Islands		
Norway	1	3
Occupied Palestinian Territory	3	1
Oman	2	3
Pakistan	3	1
Palau		
Panama	2	3
Papua New Guinea	2.5	1.5
Paraguay	2	3
Peru	2.5	2
Philippines	2.5	1.5
Pitcairn Islands		
Poland	2	3
Portugal	1	3
Puerto Rico	2	2
Qatar	2	3
Republic of Korea	1.5	3
Republic of Moldova	3	2.5
Réunion	3	2
Romania	2	3
Russian Federation	3	2

Country	PL	AP
Rwanda	2	2
Saint Helena	3	1
Saint Kitts and Nevis	2	3
Saint Lucia	2	3
Saint Pierre and Miquelon	1	3
Saint Vincent and the Grenadines	2	3
Samoa	2	3
San Marino	2	3
Sao Tome and Principe	3	2
Saudi Arabia	2	3
Senegal	2	1
Serbia	2	3
Serbia and Montenegro	2	3
Seychelles	3	2
Sierra Leone	3	1
Singapore	2.5	2.5
Slovakia	2	3
Slovenia	2	3
Solomon Islands	2	3
Somalia	3	1
South Africa	2	2
Spain	1.5	3
Sri Lanka	2.5	1
Sudan	3	1
Suriname	2.5	2
Swaziland	3	2
Sweden	1	3
Switzerland	1	3
Syrian Arab Republic	2	2
Tajikistan	3	2
Thailand	2.5	1
The former Yugoslav	2	3
Republic of Macedonia		
Timor-Leste	3	1
Togo	2.5	1
Tokelau	2	2.5
Tonga	2	3
Trinidad and Tobago	2	3
Tunisia	2	2
Turkey	2	2
Turkmenistan	3	2.5
Turks and Caicos Islands	2	3
Tuvalu	2	2.5
Uganda	2	2
Ukraine	2.5	3
United Arab Emirates	2	3
United Kingdom	1	3
United Republic of Tanzania	2	2
United States of America	1	3
United States Virgin Islands	1.5	3
Uruguay	2	3
Uzbekistan	3	2
Vanuatu	2	3
Venezuela (Bolivarian Republic of)	2	2.5
Viet Nam	3	1
Wallis and Futuna Islands	2	3
Western Sahara	2	1
Yemen	2	2
Yugoslav SFR	2	3
Zambia	3	1.5
Zimbabwe	3	1

PL = Pesticide legislation; AP = accessibility of pesticides to farmers

Table 5. Crop-specific pesticide use intensity (PUI)

Activity	PUI
Agave Fibres Nes	1
Alfalfa For Forage+Silag	0
Alfalfa Meal And Pellets	0
Almonds, with shell	2
Anise, badian, fennel, corian.	1
Apples	3
Apricots	3
Arabic Gum	0
Arecanuts	1
Artichokes	2.5
Asparagus	2.5
Avocados	3
Bambara beans	2
Bananas	3
Barley	2
Beans, dry	3
Beans, green	2
Beets For Fodder	0
Berries Nes	2
Blueberries	2
Brazil nuts, with shell	1
Broad beans, horse beans, dry	3
Buckwheat	1
Cabbages and other brassicas	3
Canary seed	1
Carobs	2
Carrots and turnips	3
Cashew nuts, with shell	1.5
Cashewapple	2
Cassava	2.5
Castor oil seed	3
Cauliflowers and broccoli	3
Cereals, most	0
Cereals, nes	2
Cherries	3
Chestnuts	1.5
Chick peas	3
Chicory roots	2.5
Chillies and peppers, dry	2.5
Chillies and peppers, green	2.5
Cinnamon (canella)	2
Citrus fruit, nes	3
Clover For Forage+Silage	0
Cloves	3

Activity	PUI
Cocoa beans	3
Coconuts	2.5
Coffee, green	3
Coir	2
Cow peas, dry	3
Cranberries	1
Cucumbers and gherkins	3
Currants	2
Dates	3
Eggplants (aubergines)	3
Eggs Excl Hen	0
Fibre Crops Nes	3
Figs	2
Flax fibre and tow	2
Fonio	1
Forage Products Nes	0
Fruit Fresh Nes	3
Fruit, tropical fresh nes	3
Fruits, most	0
Garlic	1.5
Ginger	2
Gooseberries	3
Grapefruit (inc. pomelos)	3
Grapes	3
Grass	0
Grasses Nes,Forage+Silag	0
Groundnuts, with shell	2
Gums Natural	3
Hay (Clover,Lucerne,Etc)	0
Hay (Unspecified)	0
Hay Non-Leguminous	0
Hazelnuts, with shell	1
Hemp Tow Waste	2
Hempseed	1
Hops	2
Jojoba	0
Jute	2
Kapok Fruit	2
Karite Nuts (Sheanuts)	1
Kiwi fruit	3
Kolanuts	1
Leguminous Nes,For+Sil	0
Leguminous vegetables, nes	2.5
Lemons and limes	3

Activity	PUI
Lentils	1
Lettuce and chicory	2.5
Linseed	1
Lupins	1
Maize	3
Maize For Forage+Silage	0
Maize, green	3
Mangoes, mangosteens, guavas	3
Manila Fibre (Abaca)	2
Maple	0
Maté	2
Melonseed	1
Millet	2.5
Mixed grain	2
Mules	0
Mushrooms and truffles	0
Mustard seed	1
Natural rubber	3
Nutmeg, mace and cardamoms	2
Nuts, nes	1.5
Oats	1
Oil Of Citronella	0
Oil palm fruit	3
Oils Marine Animals	0
Oilseeds, Nes	2.5
Okra	2
Olives	2
Onions (inc. shallots), green	2
Onions, dry	2
Oranges	3
Other Bastfibres	2
Other melons (inc.cantaloupes)	2.5
Papayas	3
Peaches and nectarines	3
Pears	3
Peas, dry	3
Peas, green	3
Pepper (Piper spp.)	2
Peppermint	1
Persimmons	2
Pigeon peas	2
Pineapples	2.5

Activity	PUI
Pistachios	1
Plantains	3
Plums and sloes	3
Pome Fruit Nes	3
Popcorn	2.5
Poppy seed	1
Potatoes	3
Pulses, nes	2
Pumpkins, squash and gourds	3
Pyrethrum,Dried	0
Quinces	3
Quinoa	1
Ramie	1
Rapeseed	3
Raspberries	3
Rice, paddy	3
Roots and Tubers, most	0
Roots and Tubers, nes	2.5
Rye	1
Safflower seed	2
Seed cotton	3
Sesame seed	2
Sisal	2
Sorghum	2.5
Sour cherries	3
Soybeans	3
Spices, nes	2
Spinach	3
Starch and Sugar crops for Alc	0
Stone fruit, nes	3
Straw, Husks	0
Strawberries	3
String beans	3
Sugar beet	2.5
Sugar cane	3
Sugar crops, nes	2.5
Sunflower seed	1
Swedes For Fodder	0
Sweet potatoes	2.5
Tallow tree	0
Tangerines, mandarins, clem.	3
Taro (cocoyam)	2.5
Tea	3
Tea Nes	0
Tobacco, unmanufactured	3
Tomatoes	3
Triticale	2
Tung Nuts	1
Turkeys	0
Turnips For Fodder	0
Vanilla	2
Vegetables fresh nes	2.5
Vegetables, most	0
Vegetables+Roots,Fodder	0
Vetches	1
Walnuts, with shell	1.5
Watermelons	3
Wheat	2
Yams	2.5
Yautia (cocoyam)	2.5

PUI = Pesticide use intensity; nes = other, not elsewhere specified

3.20 Soil erosion data

This section describes the soil erosion indicator used in SOLm (cf. section 3.1.38). It is based on the supplementary information of (Schader, Muller et al. 2015), i.e. the following **Fehler! Verweisquelle konnte nicht gefunden werden.**, compiled in 2013:

*Table 6. Soil erosion (from water) values in tonnes soil lost/ha*yr*

Country	min	Agricultural Land/Cropland mean	max	min	Grassland/ Pasture mean	max	Forest	Orchard	Shrubs	Vineyard	Reference
Albania	0.78		1.86								Grazhdani (2006)
Argentina	0.20	18.80	38.00		0.00						Pimentel (1993) and Lal, Hall et al. (1989) (averages)
Austria	0.50	8.93	39.00								Darmendrail, Cerdan et al. (2004) (average) and Strauss and Klaghofer (2006) (range)
Belgium	2.80	8.50	17.60								Darmendrail, Cerdan et al. (2004) (average) and Verstraeten, Poesen et al. (2006) (range)
Benin	17.00		28.00								Lal, Hall et al. (1989)
Brazil		18.80									Lal, Hall et al. (1989)
Bulgaria	0.27	4.76	5.15	0.03	2.69	6.00		12.65		12.65	Rousseva, Lazarov et al. (2006)
Burkina Faso	5.00		35.00								Pimentel (1993)
China	10.00		251.00								Lal, Hall et al. (1989)
Colombia		22.00									Lal, Hall et al. (1989)
Côte d'Ivoire	60.00		570.00								Lal, Hall et al. (1989)
Czech Republic	0.00	2.27	13.89								Dostal, Janecek et al. (2006)
Denmark	0.26	0.64	12.79		0.03						Darmendrail, Cerdan et al. (2004), Veihe and Hasholt (2006)
Ecuador	210.00		564.00								Lal, Hall et al. (1989)
Ethiopia	8.00		117.70	2.00		29.40					Taddese (2001)
Finland	0.10		2.35								Tattari and Rekolainen (2006)
France		2.03			0.01					11.09	Darmendrail, Cerdan et al. (2004)
Germany		1.32			0.14		0.00		0.13	33.23	Darmendrail, Cerdan et al.

Country	min	Agricultural Land/Cropland mean	max	min	Grassland/ Pasture mean	max	Forest	Orchard	Shrubs	Vineyard	Reference
											(2004). Auerswald (2006) (Grassland)
Ghana	5.00		10.00								Lal (1993)
Greece		0.58						0.05	1.17	0.41	Darmendrail, Cerdan et al. (2004)
Guatemala	5.00		35.00								Lal, Hall et al. (1989)
Guinea	17.90		24.50								Lal, Hall et al. (1989)
India		25.00									Ismail and Ravichandran (2008)
Indonesia (Java)	50.80		144.30								Magrath and Arens (1989)
Italy		1.33			0.28		0.20		0.06	54.86	Darmendrail, Cerdan et al. (2004)
Jamaica		90.00									Lal, Hall et al. (1989)
Kenya	25.00		45.00		5.00		1.00		7.50		Cohen, Brown et al. (2006)
Lesotho		20.00									Bojö (1996)
Lithuania	2.50	19.38	32.20		0.01						Darmendrail, Cerdan et al. (2004) (average), Jankausas and Fullen (2006) (range)
Malawi		20.00									Bojö (1996)
Mali		6.50									Bojö (1996)
Mexico	10.00		15.00								Margulis (1992)
Nepal		40.00									Lal, Hall et al. (1989)
Netherlands		6.76									Darmendrail, Cerdan et al. (2004)
Nicaragua		11.00									Alfsen, De Franco et al. (1996)
Niger	35.00		70.00								Lal, Hall et al. (1989)
Nigeria		14.40									Lal, Hall et al. (1989)
Norway	0.20		3.50	0.10		2.60					Oygarden, Lundekvam et al. (2006)
Papua New Guinea	6.00		320.00								Lal, Hall et al. (1989)
Paraguay		18.80									Lal, Hall et al. (1989)
Peru		15.00									Lal, Hall et al. (1989)

Country	min	Agricultural Land/Cropland mean	max	min	Grassland/ Pasture mean	max	Forest	Orchard	Shrubs	Vineyard	Reference
Portugal		0.59			0.04				0.40		Darmendrail, Cerdan et al. (2004)
Romania	0.70		44.80								Ionita, Radoane et al. (2006)
Russian Federation	0.50	4.80	20.00								Sidochuk, Litvin et al. (2006)
Rwanda	35.00		246.00								Berry, Olson et al. (2003)
Senegal	5.00		30.00								Pimentel (1993)
Slovakia		20.00									Stankoviansky, Fulajtar et al. (2006)
Slovenia	2.39		10.94	0.04		1.89		4.77		22.12	Hrvatin, Komac et al. (2006)
South Africa		5.00									Bojö (1996)
Spain		0.30			0.84		0.00		0.52		Darmendrail, Cerdan et al. (2004)
Switzerland		0.67									Prasuhn (2004)
The former Yugoslav Republic of Macedonia	0.04		4.77								Blinkov and Trendafilov (2006)
Turkey		2.42									Demirci and Karaburun (2012)
Uganda		5.10					0.10				Isabirye (2005)
United Kingdom	0.59	2.09	5.60		0.01						Darmendrail, Cerdan et al. (2004) Boardman and Evans (2006)
United Republic of Tanzania	10.10		92.80								Lal, Hall et al. (1989)
United States of America		6.68									Ismail and Ravichandran (2008)
Zimbabwe		43.00									Bojö [78]

4. References

Alexandratos, N. and J. Bruinsma (2012). World Agriculture Towards 2030/2050. The 2012 Revision. ESA Working Paper No. 12-03. A. D. E. Division. Rome, FAO.

Alfsen, K. H., M. A. De Franco, S. Glomsrød and T. Johnsen (1996). "The cost of soil erosion in Nicaragua." Ecological Economics **16**(2): 129-145.

Andrade de Sá, S., C. Palmer and S. di Falco (2013). "Dynamics of indirect land-use change: Empirical evidence from Brazil." Journal of Environmental Economics and Management **65**(3): 377-393.

AQUASTAT (2019). AQUASTAT database. Rome, Italy, Food and Agriculture Organization FAO.

Auerswald, K. (2006). Germany. Soil Erosion in Europe. J. Boardman and J. Poesen, John Wiley & Sons, Ltd**:** 213-230.

Badgley, C., J. Moghtader, E. Quintero, E. Zakem, M. J. Chappell, K. A. Vazquez, A. Samulon and I. Perfecto (2007). "Organic agriculture and the global food supply." Renewable Agriculture and Food Systems **22**(2): 86-108.

Berry, L., J. Olson and D. Campbell (2003). Assessing the extent, cost and impact of land degradation at the national level: findings and lessons learned from seven pilot case studies, Commissioned by Global Mechanism with support from the World BankWorld Bank.

Blinkov, I. and A. Trendafilov (2006). Macedonia. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley and Sons, Ltd.

Boardman, J. and R. Evans (2006). Britain. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Bojö, J. (1996). "Analysis- The costs of land degradation in Sub-Saharan Africa." Ecological Economics(16): 161-173.

Cohen, M. J., M. T. Brown and K. D. Shepherd (2006). "Estimating the environmental costs of soil erosion at multiple scales in Kenya using emergy synthesis." Agriculture, Ecosystems & Environment **114**(2–4): 249-269.

Darmendrail, D., O. Cerdan, Gobin A., Bouzit M., Blanchard F. and S. B. (2004). Assessing the economic impact of soil deterioration: Case Studies and Database Research., European Commission.

de Ponti, T., B. Rijk and M. K. van Ittersum (2012). "The crop yield gap between organic and conventional agriculture." Agricultural Systems **108**: 1-9.

Demirci, A. and A. Karaburun (2012). "Estimation of soil erosion using RUSLE in a GIS framework: a case study in the Buyukcekmece Lake watershed, northwest Turkey." Environmental Earth Sciences **66**(3): 903-913.

Dostal, T., M. Janecek, Z. Kliment, J. Krasa, J. Langhammer, J. Vaska and K. Vrana (2006). Czech Republic. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Erb, K. H., V. Gaube, F. Krausmann, C. Plutzar, A. Bondeau and H. Haberl (2007). "A comprehensive global 5 min resolution land-use data set for the year 2000 consistent with national census data." Journal of Land Use Science **2**(3): 191-224.

FAO (2000). Technical ConversionFactors for Agricultural Commodities. Rome, Food and Agriculture Organization of the United Nations FAO.

FAO (2001). Food Balance Sheets - A Handbook. Rome, Food and Agriculture Organization of the United Nations FAO.

FAO (2018). The future of food and agriculture – Alternative pathways to 2050. Rome, Food and Agriculture Organization of the United Nations FAO.

FAO. (2019). "FAOSTAT." 2019, from http://www.fao.org/faostat/en/#data.

FAO WFP and IFAD (2012). The State of Food Insecurity in the World 2012. Economic growth is necessary but not sufficient to accelerate reduction of hunger and malnutrition. Rome, Food and Agriculture Organization of the United Nations FAO.

Godard, C., J. Roger-Estrade, P. A. Jayet, N. Brisson and C. Le Bas (2008). "Use of available information at a European level to construct crop nitrogen response curves for the regions of the EU." Agricultural Systems **97**(1): 68-82.

Golan, A., G. Judge and D. Miller (1996). Maximum entropy econometrics. Robust estimation with limited data. Chichester, New York, Wiley.

Grazhdani, S. (2006). Albania. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Herrero, M., P. Havlik, H. Valin, A. Notenbaert, M. C. Rufino, P. K. Thornton, M. Blümmel, F. Weiss, D. Grace and M. Obersteiner (2013). "Biomass use, production, feed efficiencies, and greenhouse gas emissions from global livestock systems." Proc. Natl Acad. Sci. USA **110**(52): 20888-20893.

Herridge, D., M. Peoples and R. Boddey (2008). "Global inputs of biological nitrogen fixation in agricultural systems." Plant and Soil **311**(1-2): 1-18.

Hrvatin, M., B. Komac, D. Perko and M. Zorn (2006). Slovenia. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley and Sons, Ltd.

Ionita, I., M. Radoane and S. Mircea (2006). Romania. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

IPCC (2006). 2006 IPCC Guidelines for National Greenhouse Gas Inventories - Volume 4: Agriculture, Forestry and Other Land Use, Intergovernmental Panel on Climate Change (IPCC).

IPCC (2018). Global Warming of 1.5°C. An IPCC Special Report, Intergovernmental Panel on Climate Change IPCC.

Isabirye, M. (2005). Land Evaluation Around Lake Victoria: Environmental Implications of Land Use Change, Katholieke Universiteit Leuven.

Ismail, J. and S. Ravichandran (2008). "RUSLE2 Model Application for Soil Erosion Assessment Using Remote Sensing and GIS." Water Resources Management **22**(1): 83-102.

Jankausas, B. and M. A. Fullen (2006). Lithuania. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Kastner, T., M. Kastner and S. Nonhebel (2011). "Tracing distant environmental impacts of agricultural products from a consumer perspective." Ecological Economics **70**(6): 1032-1040.

Kissinger, G., M. Herold and V. De Sy (2012). Drivers of Deforestation and Forest Degradation: A Synthesis Report for REDD+ Policymakers. Vancouver, Canada, Lexeme Consulting.

Lal, R. (1993). Soil erosion and conservation in West Africa. World Soil Erosion and Conservation. D. Pimentel and Cambridge University Press.

Lal, R., G. F. Hall and F. P. Miller (1989). "Soil Degradaton: 1. Basic Processes." Land Degradation & Rehabilitaion **1**: 51-69.

Leifeld, J. and L. Menichetti (2018). "The underappreciated potential of peatlands in global climate change mitigation strategies." Nature Communications **9**(1): 1071.

Lu, X., H. Jiang, X. Zhang, J. Liu, Z. Zhang, J. Jin, Y. Wang, J. Xu and M. Cheng (2013). "Estimated global nitrogen deposition using NO2 column density " International Journal of Remote Sensing **34**(24): 8893-8906.

Magrath, W. and P. Arens. (1989). "The costs of soil erosion on Java: a natural resource accounting approach." Environment Department working paper ; no. ENV 18 Retrieved 13.09., 2013, from http://documents.worldbank.org/curated/en/1989/08/737809/costs-soil-erosion-java-natural-resource-accounting-approach.

Margulis, S. (1992). "Back of the envelope estimates of environmental damage costs in Mexico." Policy Research Working Paper Series(824).

Meyfroidt, P., E. F. Lambin, K.-H. Erb and T. W. Hertel (2013). "Globalization of land use: distant drivers of land change and geographic displacement of land use. Current Opinion in Environmental Sustainability, Human settlements and industrial systems " 5: 438–444.

Muller, A., C. Schader, N. El-Hage Scialabba, J. Brügemann, A. Isensee, K.-H. Erb, P. Smith, K. Klocke, F. Leiber, M. Stolze and U. Niggli (2017). "Strategies for feeding the world more sustainably with organic agriculture." Nature Communications **8**(1290).

Myhre, G., D. Shindell, F. Bréon, W. Collins, J. Fuglestvedt, J. Huang, D. Koch, J. Lamarque, D. Lee, B. Mendoza, T. Nakajima, A. Robock, G. Stephens, T. T and H. Zhang (2013). IPCC AR5, Chapter 8: Anthropogenic and Natural Radiative Forcing Climate Change 2013: The Physical Science Basis. Contribution of

Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, IPCC.

Nemecek, T., A. Heil, O. Huguenin, S. Meier, S. Erzinger, S. Blaser, D. Dux. and A. Zimmermann (2007). Life Cycle Inventories of Agricultural Production Systems. Dübendorf, CH, Agroscope FAL Reckenholz and FAT Taenikon, Swiss Centre for Life Cycle Inventories.

OECD. (2019). "Nutrient Balance." from https://data.oecd.org/agrland/nutrient-balance.htm.

Oygarden, L., H. Lundekvam, A. H. Arnoldussen and T. Borresen (2006). Norway. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Pimentel, D., ed. (1993). World Soil Erosion and Conservation, Cambridge University Press.

Ponisio, L. C., L. K. Gonigle, K. C. Mace, J. Palomino, P. de Valpine and C. Kremen (2014). "Diversification practices reduce organic to conventional yield gap." Proceedings of the Royal Society of London B: Biological Sciences **282**(1799).

Prasuhn, V. (2004). Mapping of actual soil erosion in Switzerland. Agroscope Reckenholz, Zürich-Reckenholz, Switzerland, Swiss Federal Research Station for Agroecology and Agriculture.

Ramaswamy, V., O. Boucher, J. Haigh, D. Hauglustaine, J. Haywood, G. Myhre, T. Nakajima, G. Shi and S. Solomon (2001). IPCC TAR, Chapter 6: Radiative Forcing of Climate Change. Climate Change 2001: The Physical Science Basis. Contribution of Working Group I to the THird Assessment Report of the Intergovernmental Panel on Climate Change, IPCC.

Rousseva, S., A. Lazarov, E. Tsvetkova, I. Marinov, I. Malinov, V. Kroumov and V. Stefanova (2006). Bulgaria. Soil Erosion in Europe.

J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Schader, C., A. Muller, N. El-Hage Scialabba, J. Hecht, A. Isensee, K.-H. Erb, P. Smith, H. Makkar, P. Klocke, F. Leiber, P. Schwegler, M. Stolze and U. Niggli (2015). "Impacts of feeding less food-competing feedstuffs to livestock on global food system sustainability." Journal of the Royal Society Interface **12**: 20150891.

Seufert, V. (2018). Comparing Yields: Organic Versus Conventional Agriculture. Encyclopedia of Food Security and Sustainability. P. Ferranti, E. M. Berry and J. R. Anderson. Oxford, Elsevier: 196-208.

Seufert, V., N. Ramankutty and J. A. Foely (2012). "Comparing the yields of organic and conventional agriculture." Nature **doi:10.1038/nature11069**.

Sidochuk, A., L. Litvin, V. Golosv and A. Chernysh (2006). European Russia and Byelorus. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley and Sons, Ltd.

Smith, P., P. Gregory, D. van Vuuren, M. Obersteiner, M. Rounsevell, J. Woods, P. Havlik, E. Stehfest and J. Bellarby (2010). "Competition for land." Philosophical Transactions of the Royal Society, B **365**: 2941-2957.

Stankoviansky, M., E. Fulajtar and P. Jambor (2006). Slovakia. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Strauss, P. and E. Klaghofer (2006). Austria. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Taddese, G. (2001). "Land Degradation: A Challenge to Ethiopia." Environmental Management **27**(6): 815-824.

Tattari, S. and S. Rekolainen (2006). Finland. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Tubiello, F., M. Salvatore, S. Rossi, A. Ferrara, N. Fitton and P. Smith (2013). "The FAOSTAT database of greenhouse gas emissions from agriculture." Environmental Research Letters **8**(015009): 10.

UNFCCC. (2019). "GHG data from UNFCCC." from https://unfccc.int/process-and-meetings/transparency-and-reporting/greenhouse-gas-data/ghg-data-unfccc/ghg-data-from-unfccc.

Veihe, A. and B. Hasholt (2006). Denmark. Soil Erosion in Europe. J. Boardman and J. Poesen. West Sussex, England, John Wiley & Sons, Ltd.

Verstraeten, G., J. Poesen, D. Goossens, K. Gillijns, C. Bielders, D. Gabriels, G. Ruysschaert, M. Van Den Eeckhaut, T. Vanwalleghem and G. Govers (2006). Belgium. Soil Erosion in Europe. J. Boardman and J. Poesen, John Wiley & Sons, Ltd**:** 385-411.

Walpole, S. C., D. Prieto-Merino, P. Edwards, J. Cleland, G. Stevens and I. Roberts (2012). "The weight of nations: an estimation of adult human biomass." BMC Public Health **12**: 439.

Water Footprint Network (2019). Water Footprint Network. Enschede, The Netherlands, Water Footrpint Network.

Wood, S. and A. Cowie (2004). A Review of Greenhouse Gas Emission Factors for Fertiliser Production, IEA Bioenergy Task 38.