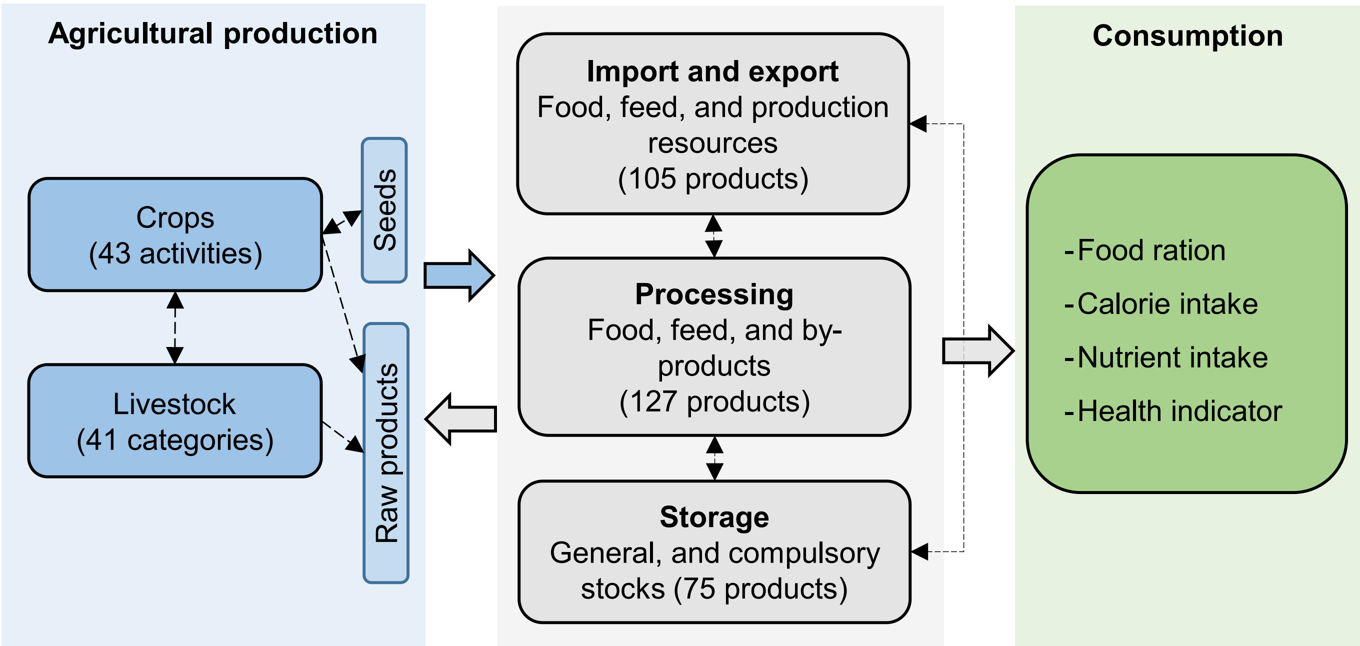
Agroscope Science | Nr. 214 / 2025



Documentation of the Swiss Sustainable Food System (SWISSfoodSys) model

Version 1.0

Autoren

Utkur Djanibekov, Albert von Ow

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
Agroscope

Imprint

Publisher	Agroscope Tänikon 1 8356 Ettenhausen www.agroscope.ch
Information	Djanibekov Utkur utkur.djanibekov@agroscope.admin.ch
Layout	Jacqueline Gabriel
Cover Figure	Albert von Ow, Utkur Djanibekov
Download	www.agroscope.ch/science
Copyright	© Agroscope 2025
ISSN	2296-729X
DOI	https://doi.org/10.34776/as214e

Table of Contents

Abstract

This documentation describes the Swiss Sustainable Food System (SWISSfoodSys) model. We describe in detail the sets, parameters, variables, and equations of the model and their programming codes written using the General Algebraic Modeling System (GAMS); we present the state of the model until April 2025. The model aims to improve the understanding of the impacts of political, technological, socioeconomic, and environmental changes on the Swiss food system for the period between 2019 and 2050. To analyze the possible impacts of these changes, the model is ex-ante and simulates “what-if” scenarios. The SWISSfoodSys model consists of crop production, livestock rearing and feeding, storage of products, processing of raw agricultural products, trade of commodities, environmental impacts, agricultural incomes, employment, domestic food self-sufficiency and food consumption modules. Different components of the model can be extended with additional equations or, instead, can be deactivated, for example, storage and employment modules can be omitted in certain scenarios. To address the temporal relationships among the modules, the model utilizes a dynamic programming approach. The model can be adjusted to recursive programming to assess the sequential impacts of changes and other programming approaches, depending on modeling requirements. A large number of equations of the SWISSfoodSys model are based on the Decision Support System – Ernährungssicherungsstrategie Angebotslenkung (DSS-ESSA) model. DSS-ESSA was used to investigate the impacts of preventing the food supply crisis and optimize agricultural production to have the best possible food supply in Switzerland under such situations. We also describe the transfer and validation of these equations into the SWISSfoodSys model. The model outputs can be used by policymakers (e.g., Federal Office of Agriculture) to understand the impacts of potential new policies and accordingly better design these policies. The model can also be utilized by researchers to investigate changes to the food system under different scenarios. The model documentation is intended to be updated regularly.

Resumé

La présente documentation décrit le modèle SWISSfoodSys (Swiss Sustainable Food System). Elle fournit une vue d’ensemble complète du modèle, incluant l’ensemble des séries, paramètres, variables et équations, ainsi que les codes de programmation développés dans le langage GAMS (General Algebraic Modeling System). La version du modèle documentée ici correspond à son état en avril 2025. Le modèle vise à mieux comprendre l'impact des changements politiques, technologiques, socio-économiques et environnementaux sur le système alimentaire suisse sur la période 2019 - 2050. Dans cette optique, le modèle SWISSfoodSys adopte une approche ex ante, reposant sur la simulation de différents scénarios hypothétiques. Le modèle SWISSfoodSys se compose des modules suivants: production végétale, détention et alimentation animales, stockage, transformation des matières premières agricoles, commerce extérieur des denrées alimentaires et des aliments pour animaux, impact environnemental, revenus agricoles, emploi, taux d'auto-approvisionnement et consommation alimentaire. Certains modules peuvent être désactivés ou complétés par des équations spécifiques selon les scénarios analysés. Les modules stockage et emploi peuvent par exemple être omis dans certaines configurations. Pour représenter les relations dans le temps, le modèle utilise une approche de programmation dynamique. Il peut également être adapté à une programmation récursive pour évaluer les effets séquentiels des changements, ainsi qu'à d'autres approches de programmation en fonction des besoins de modélisation. Une grande partie des équations du modèle SWISSfoodSys s’appuie sur celles du modèle DSS-ESSA (Decision Support System - Stratégie de sécurité alimentaire - Gestion de l'offre). Le DSS-ESSA a permis d'étudier quelles seraient les adaptations nécessaires en termes de production pour garantir au mieux l'approvisionnement alimentaire en Suisse en situation de crise. Nous décrivons également le transfert et la validation des équations issues de DSS-ESSA dans le modèle SWISSfoodSys. Les résultats du modèle offrent un appui précieux aux décideurs politiques (notamment à l'Office fédéral de l'agriculture) en les aidant à évaluer l'impact potentiel de nouvelles mesures politiques et à les adapter en conséquence. Le modèle peut également être utilisé par les chercheuses et chercheurs pour explorer les évolutions possibles du système alimentaire suisse à travers divers scénarios. La documentation du modèle est appelée à être régulièrement actualisée.

Zusammenfassung

Die vorliegende Dokumentation beschreibt das Modell SWISSfoodSys (Swiss Sustainable Food System). Die Beschreibung umfasst alle Sets, Parameter, Variablen und Gleichungen des Modells und die Programmiercodes, die mit dem General Algebraic Modeling System (GAMS) geschrieben wurden. Der Stand des dokumentierten Modells ist April 2025. Mit dem Modell soll das Verständnis der Auswirkungen politischer, technologischer, sozioökonomischer und ökologischer Veränderungen auf das Schweizer Ernährungssystem für den Zeitraum zwischen 2019 und 2050 verbessert werden. Um die möglichen Auswirkungen solcher Veränderungen zu analysieren, verfolgt das Modell einen Ex-ante-Ansatz und simuliert Was-wäre-wenn-Szenarien. Das Modell SWISSfoodSys besteht aus den Modulen Pflanzenbau, Tierhaltung und Fütterung, Lagerhaltung, Verarbeitung von landwirtschaftlichen Rohstoffen, Aussenhandel mit Nahrungs- und Futtermitteln, Umweltauswirkungen, landwirtschaftliche Einkommen, Beschäftigung, Selbstversorgungsgrad und Nahrungsmittelkonsum. Verschiedene Komponenten des Modells können deaktiviert oder um zusätzliche Gleichungen erweitert werden, z. B. können die Module Lagerung und Beschäftigung in bestimmten Szenarien weggelassen werden. Um die zeitlichen Beziehungen zu berücksichtigen, verwendet das Modell einen Ansatz der dynamischen Programmierung. Das Modell kann für die Bewertung sequenzieller Auswirkungen von Änderungen an die rekursive Programmierung angepasst werden, ebenso wie an weitere Programmieransätze je nach Modellierungsanforderungen. Ein Grossteil der Gleichungen des Modells SWISSfoodSys basiert auf dem Modell DSS-ESSA (Decision Support System – Ernährungssicherungs-strategie Angebotslenkung). Mit DSS-ESSA wurde untersucht, welche Produktionsanpassungen erforderlich sind, um die Nahrungsmittelversorgung in der Schweiz in einer Nahrungsmittelkrise bestmöglich zu gewährleisten. Wir beschreiben auch die Übertragung und Validierung dieser Gleichungen in das Modell SWISSfoodSys. Die Modellergebnisse helfen politischen Entscheidungsträgern (z. B. dem Bundesamt für Landwirtschaft), die Aus-wirkungen möglicher neuer politischer Massnahmen zu verstehen und diese entsprechend anzupassen. Das Modell kann auch von Forschenden dazu genutzt werden, Veränderungen im Lebensmittelsystem durch verschiedene Szenarien zu untersuchen. Es ist vorgesehen, dass die Dokumentation des Modells regelmässig aktualisiert wird.

Riassunto

Il presente documento descrive il modello SWISSfoodSys (Swiss Sustainable Food System). Fornisce una visione dettagliata degli insiemi, i parametri, le variabili e le equazioni del modello e i relativi codici di programmazione scritti con il General Algebraic Modeling System (GAMS) e viene qui presentato nel suo stato ad aprile 2025. Il modello punta a offrire una migliore visione dell’impatto dei cambiamenti politici, tecnologici, socioeconomici e ambientali sul sistema alimentare svizzero per il periodo compreso tra il 2019 e il 2050. In quest’ottica, adotta un approccio ex-ante e si basa sulla simulazione di diversi scenari ipotetici. Il modello SWISSfoodSys è composto dai seguenti moduli: produzione vegetale, detenzione e alimentazione animale, stoccaggio di prodotti, trasformazione delle materie prime agricole, commercio delle materie prime, impatto ambientale, reddito agricolo, occupazione, grado di autosufficienza nazionale e consumo alimentare. I diversi componenti del modello possono essere disattivati o completati con equazioni aggiuntive; in determinati scenari è ad esempio possibile omettere i moduli relativi a stoccaggio e occupazione. Il modello utilizza un approccio di programmazione dinamica per rappresentare le relazioni temporali tra moduli. Può inoltre essere adattato alla programmazione ricorsiva per valutare gli impatti sequenziali dei cambiamenti e ad altri approcci di programmazione, a seconda delle esigenze di modellazione. Un gran numero di equazioni del modello SWISSfoodSys si basa sul modello Decision Support System - Strategia di sicurezza alimentare Gesione dell’offerta (DSS-ESSA). Il DSS-ESSA ha permesso di studiare gli adattamenti necessari in termini di produzione per garantire al meglio l’approvvigionamento alimentare in Svizzera in situazioni di crisi. Descriviamo inoltre il trasferimento e la verifica di tali equazioni nel modello SWISSfoodSys. I risultati del modello rappresentano un prezioso spunto per le legislatrici e i legislatori (ad esempio per l'Ufficio federale dell'agricoltura), permettendo loro di valutare l’impatto di potenziali nuove politiche e di adattarle di conseguenza. Il modello può essere utilizzato anche da ricercatrici e ricercatori per studiare i possibili cambiamenti del sistema alimentare attraverso diversi scenari. La documentazione del modello sarà aggiornata regolarmente.

1 Introduction

This documentation describes an economic mathematical programming model for a sustainable food system in Switzerland. This includes a documentation and detailed description of the Swiss Sustainable Food System (SWISSfoodSys) model, including its equations, variables, sets, and parameters. The model takes into account the specific characteristics of the food system of Switzerland. The SWISSfoodSys model analyzes the physical, financial, and environmental relationships involved in the food system. Further, the model mainly aims to analyze the impact of political, technological, socioeconomic, and environmental changes on the food system. It consists of several modeling components, including agricultural production (crop production and livestock rearing), livestock feeding, storage of products, processing, trade, environmental impacts, agricultural incomes, employment, domestic food self-sufficiency, and food consumption. The model can support policymakers on how changing framework conditions, such as improving consumers' well-being (e.g., food security, improving nutrition and healthy diets), environmental management (e.g., reductions in greenhouse gases (GHGs), increase in biodiversity), agricultural incomes (e.g., increase in incomes of the agricultural sector), food self-sufficiency (e.g., increase of domestic agricultural production), and sustainable agricultural production (e.g., organic agricultural production) affect the Swiss food system, from production to consumption.

The current version of the model (status as of April 2025) employs a deterministic and dynamic linear programming approach. The model can be further extended by including other modeling components, relevant equations, variables, and parameters. For example, the model can be extended by including technological, income, and added value at the production, processing, and trade stages, and food demand functions of consumers. We wrote the model using the General Algebraic Modeling System (GAMS) software. The model, which was programmed using GAMS software, features clearly defined inputs and outputs to extend the model and to replicate it in other interfaces.

A large number of equations of the model (and its programming codes) are transferred into GAMS software from the initial version of the model called Decision Support System-Ernährungssicherungsstrategie Angebotslenkung (DSS-ESSA), which was written using Linear Programming Language (LPL) software (von Ow, 2019). DSS-ESSA was used to analyze the prevention of a food supply crisis and optimize the required crop areas and livestock numbers to have the best possible food supply for Swiss population under the available production resources in crisis situations. We validated the performance of the model programmed in GAMS software by comparing its results with the model results programmed in LPL (see section 7).

In the following sections, we first describe the overall structure of the model. Thereafter, we describe the modelling components and their programming codes in GAMS software.

2 Overall description of the model

2.1 Structure of the SWISSfoodSys model

The SWISSfoodSys model analyses the Swiss food system from production to consumption. Therefore, the model includes the main agricultural production activities of the Swiss food system (Figure 1). The raw products resulting from agricultural production can then be used at the processing stage, where the raw agricultural products are processed and can be used as food, feed for livestock, seeds for crop production, and stored for subsequent use, or traded internationally. The trade includes export and import of products, beginning from the agricultural and processing stages. Imports of products consist of both raw and processed agricultural products and are used as food for human consumption or as feed for livestock. Imported products can also be stored and processed. Consumers use processed products to meet their food calorie consumption requirements. Consumption of food affects the nutrient intake level and health of consumers.

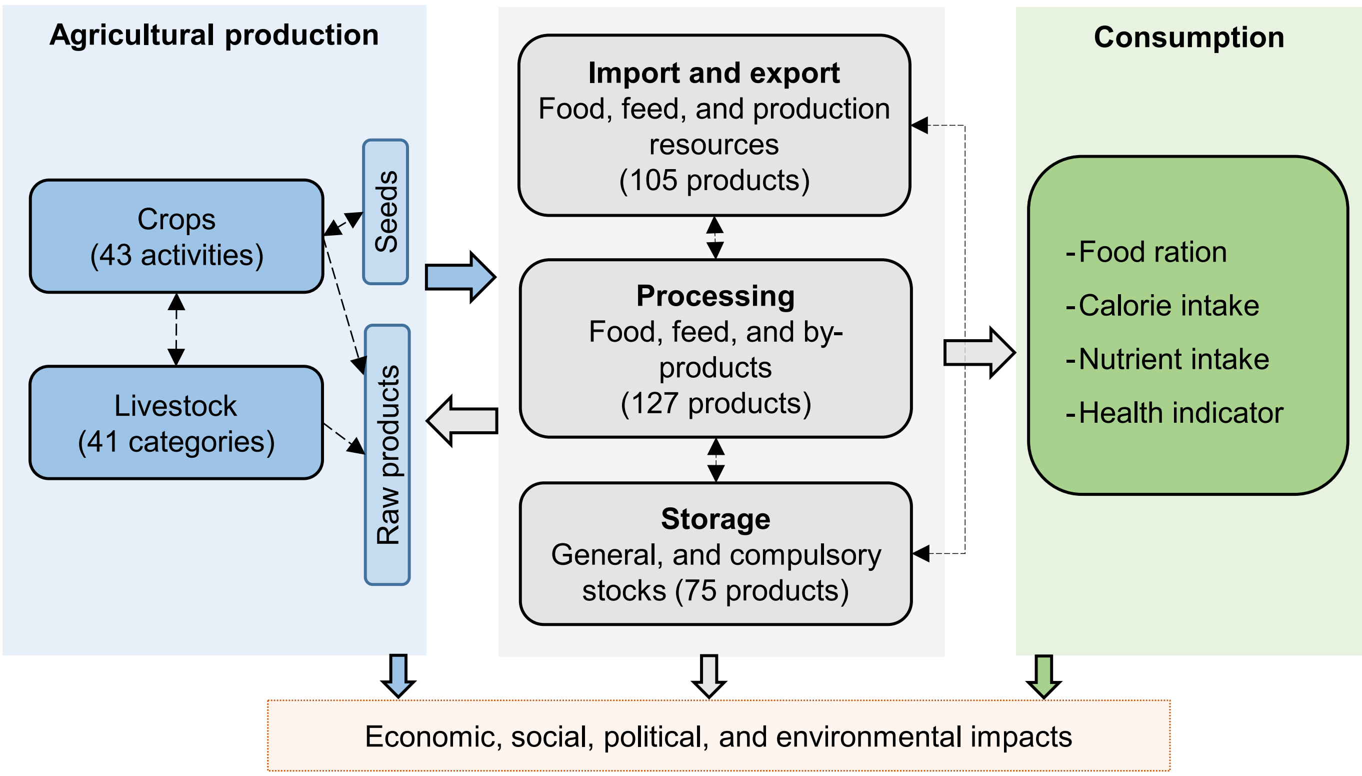


Figure 1: The Structure of the Swiss Sustainable Food System (SWISSfoodSys) model. Note: Adapted from von Ow et al. (2020).

Figure 2 provides an example of the flow of products from dairy cows, which represent agricultural production, to the consumption stage. The model accounts for 14 food products for human consumption and three feed products for livestock that originate from the production of raw milk, beef meat, animal fat, offal, and bones at the agricultural stage. At the consumption stage, we calculate the intake of nutrients and calories, and the change in health of a consumer. The appendix provides examples of the flow of products from wheat, apples, and pigs. Depending on data availability, we also include as many products as possible and their impacts in each stage of the food system.

The model produces the economic, social, political, and environmental impacts (Table 1). Economic impacts include agricultural incomes, production costs, and quantities. Social impacts include a balanced diet according to food recommendations and farm employment. Environmental impacts include GHG emissions, nutrient balances, pesticide risk indicators, and other environmental indicators. Environmental impacts are calculated for agricultural production, processing, import, export, and food cooking. Political impacts include feed concentrate production and import, food waste from production, processing, trade and consumption, and food supply self-sufficiency.

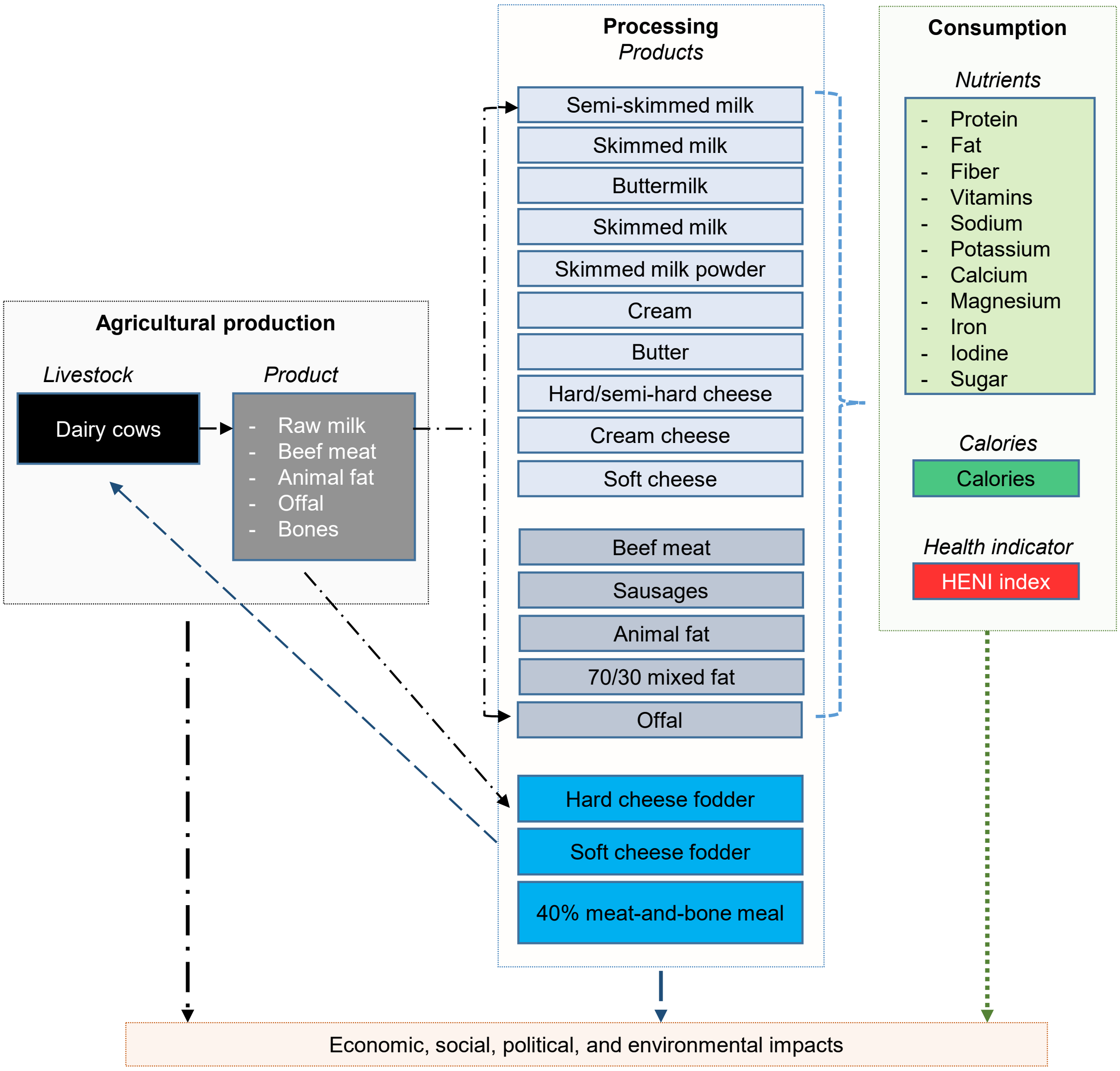


Figure 2: Example of flow of products from dairy cows in the agricultural production, processing, and consumption stages and their impacts.

Table 1: Examples of model impacts.

Impact domains	Indicators
Economic	Agricultural incomes, in CHF
	Agricultural direct payments, in CHF
	Agricultural employment, in working days
Social	Health impacts of food consumption, in HENI index
	Nutritious food consumption, in NRF10.3 index
	Diet according to Swiss food pyramid, in food portions
Political	Feed concentrate production and import, in tones
	Food waste from production, processing, trade and consumption, in tons
	Food supply self-sufficiency, in kcal
Environmental	GHG emissions, in tCO_2e
	Biodiversity, in hectares
	Nitrogen emissions, in tons of nitrogen
	Pesticide risks, in pesticide risk indicator

2.2 Dynamic character of the model

The model relies on dynamic and linear programming approaches and includes linear relationships of variables and equations. Depending on the aim of the model, the objective function either maximizes or minimizes the objective value (see section 5). The model includes several time periods, which currently consist of annual periods ranging from 2019 to 2050. We consider 2019 as the initial year of the model to exclude the temporary impacts of COVID-19 on the food system. Depending on the research questions, the timeframe of the model can be divided into monthly periods or aggregated into several years.

The model optimizes the objective function over time in a dynamic setting. Connecting different modules over time enables interperiod economic, social, political, and environmental path dependencies. The periods of the model are linked through various variables, including agricultural area in hectares, livestock number in livestock units and production in tons, and commodity storage in tons, among others. The model system takes into account the initial state (e.g., the stocks and crops already sown that were present in the observed initial year of 2019) as well as the development of the model variables over time (e.g., the rearing of young cattle with the transfer of stocks from one age group to the next age group).

We model the interperiod relationships using a dynamic programming approach. The dynamic programming optimizes the objective value throughout the entire projected period—that is, all the modeled periods are considered where the objective function sums the values of all periods in the optimization process and yields the optimal outcome for the analyzed timeframe (i.e., optimal output in the 2019–2050 period) and not for individual years (i.e., model results may not be optimal for an individual year such as 2019, 2020, until 2050). Therefore, the resulting optimal value of the model is for the entire simulated period. The model results can also be presented for each period or for specified timeframes. In the dynamic programming model, the decisions made at a certain point influence the decisions before and after that point. In such a modelling approach, the optimal decisions over time are not simulated from year to year in a recursive manner, where the optimal values are calculated for each year. However, depending on the modelled food system settings, the model can be changed to a recursive dynamic programming model where each modelled period is optimized.

3 Definitions of sets, parameters, and variables

3.1 Sets

Sets are collections of elements that are used in parameters, variables, equations, conditions and other programming operations in GAMS. We include sets of the model in the GAMS file *Sets.gms*. In this GAMS file, first the name of the sets is listed; the descriptions of the sets in German are given next to the names and the names in English are given in parentheses. To close the command that includes the list of sets, a semicolon *(";")* command is entered. Figure 3 provides an example of the sets included in GAMS.

```
SETS
eg                 Erschwerniszone (regions)
monat              Monate (months)
ist                Infrastruktur (infrastructure)
inte               Intensitaet Produktionsmethoden (production methods)
ek                 Einzelkulturen (single crops)
kg                 Kulturgruppenn (crop groups)
tk                 Tierkategorien (animal category)
vg(tk)             Vorgeneration (previous generation animals)
fg(tk)             Folgegeneration (next generation animals)
mt(tk)             Muttertiere (mother animal dams)
nk(tk)             Nachkommen (litters)
nt(tk)             Nutztiere (farm animals)
th                 Tierhaltungen (animal husbandry or animal farms)
pr                 Produkte (products)
rp(pr)             Rohprodukte (raw products)
prp(rp)            Pflanzliche Rohprodukte (plant raw products)
trp(pr)            Tierische Rohprodukte (animal raw products)
sg(rp)             Saatgut (seeds)
fm(pr)             Futtermittel (fodder)
;
```

Figure 3: Example of sets listed in Sets.gms.

The elements of the sets are defined in an Excel file titled *Setnames.xlsx*. The elements of sets are called from *Setnames.xlsx* file by using the GDX command. In Figure 4, we present an example of the elements of the sets *eg* (three regions: mountain, hill, and valley) and *ek* (crops) imported from Excel files into GAMS files. With the command *$onecho* > *sets.txt*, we send multiple sets to an external txt file called *sets.txt*.

In the next rows, we define the sets to be imported. In each row, we first indicate that the import is related to a set by including the command set and type that it is equal ("=") to the certain set (in our case these are *eg* or *ek*). Thereafter, we indicate the range of Excel cells from where the set elements are to be imported using the command *rng* (i.e., *a2:a4* cell range for elements of set *eg* and *a2:a44* cell range for elements of set *ek*), and the dimension of a set (one dimension for both sets *eg* and *ek* (*rdim = 1*)). Similarly, other sets are indicated between *$onecho* and *$offecho* commands. The action of listing sets is stopped with the *$offecho* command.

In the next row, using *gdxxrw* utility, GAMS reads sets from the Excel file called *Setnames.xlsx* and writes sets to a GDX file called *Setnames.gdx*. The *$gdxin* command is used in a sequence to load *setnames.gdx* file. The *$load* command loads specified items from the GDX file (in our case elements of the sets *eg* and *ek*) into the file *Set.gms*. Similarly, the names of other sets are indicated in the same line as the *$load* command. Table 2 provides the complete list and description of sets.

```
$onecho > sets.txt
set=eg          rng=eg!a2:a4            rdim=1
set=ek          rng=ek!a2:a44           rdim=1
$offecho
$call  gdxxrw  setnames.xlsx  trace=3 @sets.txt
$gdxin  setnames.gdx
$load eg ek
$gdxin
```

Figure 4: Example of the import of sets using the GDX file.

Table 2: List of sets used in the model.

Set name	Member (subset) of a set	Description
Regions and infrastructure		
eg		Production regions: valley, hill, mountain
ist		Four types of infrastructure objects
Periods		
pp		Periods including regeneration after crisis, currently in years
p(pp)	pp	Periods including period of crisis, currently in years
monat		12 months
Crops		
ek		43 single crops
inte		Production methods: conventional, extenso, organic
kg		20 crop groups
kgff(kg)	kg	13 crop groups with crop rotation restriction
EkZuKg(ek,kg)	ek, kg	Allocation of crops to crop groups
ek_int_eg(ek,inte,eg)	ek, inte, eg	Allocation of crops by production methods and regions
ek_int_eg_prp(ek,inte,eg,prp)	ek, inte, eg, prp	Allocation of raw crop products by crop type, production methods and regions
ek_sg(ek,sg)	ek, sg	Assignment of seed products to crops including 11 seeds used for (or produced from) 12 crops
Livestock		
tk		41 livestock categories
vg(tk)	tk	39 previous generation livestock categories
fg(tk)	tk	26 next generation livestock categories
mt(tk)	tk	12 mother livestock categories
nk(tk)	tk	15 offspring livestock categories
nt(tk)	tk	20 farmed livestock categories (without a next livestock category)
th		6 livestock groups
TkZuTh(tk,th)	tk, th	41 livestock categories by livestock groups
Tt(vg, fg)	vg, fg	Assignment of previous to next livestock category including replacement of 15 previous generation livestock categories by 26 next generation livestock categories
rr(mt,nk)	mt, nk	Assignment of offspring to mother livestock category including reproduction by 12 mother livestock categories of 15 offspring livestock categories

Set name	Member (subset) of a set	Description
w		Number of offsprings that can be borne by livestock (maximum of 365 offsprings, one day for each livestock offspring)
Feed		
Fm(pr)	pr	55 feed products
fw		13 types of feed nutrients/ingredients
Fmg		5 feed groups
bf(fm)	fm	19 domestically produced feed by-products
tk_fw(tk,fw)	tk,fw	13 feed nutrient requirements by livestock categories
Ft(fm,tk,fmg)	fm, tk, fmg	Consumption of 55 feed products that are grouped into 5 feed groups by 41 livestock categories (grouping of feed products to feed groups can differ among livestock categories)
FmgJeTk(tk,fmg)	tk, fmg	Five feed groups assigned to livestock categories
mnp(bf)	bf	Grain fodder by-products used to restrict concentrated feed consumption by livestock
oel(bf)	bf	Oil fodder by-products used in restricting concentrated feed consumption by livestock
Products		
pr		181 products
pt		3 product categories (raw, food, feed)
sp		10 other raw products (raw products that are not domestically produced or that are modelled as constantly supplied)
rp(pr)	pr	50 raw products
prp(rp)	rp	33 raw plant products
ap(pr)	pr	46 agricultural raw products that are used to produce food and feed products (raw products used for seed are not included in this set)
trp(pr)	pr	12 raw livestock products
tk_tp(tk,trp)	tk, trp	Two raw products (raw milk, eggs) produced by livestock categories
tk_sp(tk,trp)	tk, trp	10 products produced from slaughtered livestock categories
sg(rp)	rp	13 seed products
nm(pr)	pr	76 food products
vp(pr)	pr	92 domestically processed (main) food and feed products
kp(pr)	pr	22 domestically processed food and feed by-products
fp(pr)	pr	Seven products that are available at a fixed amount, which is not modelled by domestic production or import
hp(pr)	pr	105 foreign trade products including raw, food and feed products
lp(pr)	pr	75 products that can be stored including raw, food and feed products
plp(pr)	pr	Six compulsory storage products, including raw, food, and feed products

Set name	Member (subset) of a set	Description
vb(ap,vp)	ap, vp	117 processed products (food and feed) from raw products (crop and livestock). This assignment defines the process tha can result in an additional supply of one or more by-products (see vbk)
vbk(ap,vp,kp)	ap, vp, kp	Assignment of by-products to a process in which a main product is produced
VerarbKapazitaeten(ap,vp,ist)	ap, vp, ist	Processing infrastructure for processed products
Lagerkapazitaeten(lp,ist)	lp, ist	Storage infrastructure for stored products
Nutrition		
nst		Three macronutrients + alcohol
nms		13 micronutrients
nmg		32 food groups based on the food pyramid food groups
Environment		
uw		101 environmental impacts
uwa(uw)	uw	Six selected environmental impacts

3.2 Parameters

The parameters of the model include data inputs for the model. Most of the parameters are specified in the Excel file. A few of the parameters are directly calculated in the model. The model parameters in GAMS are specified in the *Parameters.gms* file. In this GAMS file, first, the command *Parameters* is indicated. After the command *Parameters*, the name of the parameters with their set domains are listed. For a better definition of parameters in programming codes, the name of every parameter begins with *"p_"* and then the unique name of the parameter is given. Next to the names of parameters, their description in German is given and that in English is given in parentheses. To close the command that includes the list of parameters, a semicolon *(";")* command is entered. Figure 5 provides an example of the parameters included in GAMS.

```
Parameters
p_DauerP(pp)                 Zeitdauer (time periods)
p_AktFlaeche(ek,inte,eg)     Aktuelle flaeche ha
*                             (current land use area in hectares)
p_Bedarf_SG(ek,sg)           Saatgutbedarf kg per ha (seed needs)
p_ErsterErntetag(ek)         ErsterErntetag Tag (first harvest day for crop)
p_Ertrag(ek,inte,eg,prp)     Ertrag je pflanzliches Rohprodukt t per ha
*                           (yield per raw crop product)
p_AktuelGemFla(ek)           Aktuellegemüseflaechen ha (actual area of vegetables)
;
```

Figure 5: Example of the parameters listed in Parameters.gms.

Data for the parameters are defined in the Excel file under the name *Data.xlsx*. This data are imported from *Data.xlsx* file using the GDX command.

Figure 6 presents an example of importing data *p_Tage* (number of days in each year) and *p_AktFlaeche* (current crop area) from the Excel file into GAMS. With the command *$onecho > ModelParameter.txt*, we send multiple listed parameters to an external txt file called *Modelparameter.txt*. In the next rows, we define the parameters that need to be imported. In each row, we first indicate that import is related to a parameter by including a command *par* and then type that it is equal to ("=") the specific parameter (in our case, these are parameters *p_Tage* or *p_ AktFlaeche*). Thereafter, we indicate the range of Excel cells from where the parameter data are imported using the command *rng* (i.e., *a2:b38* cell range for data for the parameter *p_Tage* and *a2:d297* cell range for data for the parameter *p_AktFlaeche*), and the dimension of a parameter (which is one-dimensional data for parameter *p_ Tage* (*rdim = 1*) and three-dimensional data the for the parameter *p_AktFlaeche (rdim = 3)*). Similarly, other parameters are indicated between the *$onecho* and *$offecho* commands. The action is ceased with the *$offecho* command.

In the next row, using *gdxxrw* utility, GAMS reads the parameters from *Data.xlsx* Excel file and writes the parameter data to a GDX file called *Data.gdx*. The *$gdxin* command is used in a sequence to load a *Data.gdx* file. The *$load* command loads specified items from the GDX file (in our case, data for the parameters *p_ Tage* and *p_AktFlaeche*) into the file *Parameters.gms*. Similarly, most of the data for other parameters are indicated in the same line as the *$load* command. Some of the data are not imported from the Excel file; instead, they are calculated in *Parameters.gms*. Table 3 provides the complete list and description of parameters.

```
$onecho > ModelParameter.txt
par=p_Tage          rng=Tage!a2:b38          rdim=1   cdim=0
par=p_AktFlaeche    rng=AktFlaeche!A2:d297    rdim=3   cdim=0
$offecho
$call gdxxrw  Data.xlsx  trace=3  @ModelParameter.txt
$gdxin  Data.gdx
$loaddc p_Tage p_AktFlaeche
$gdxin
```

Figure 6: Example of the import of parameters using the GDX file.

Table 3: List of all parameters utilized in the model.

Parameter name	Unit	Description
Miscellaneous		
p_Tage(pp)	1–365	Number of days
p_DauerP(pp)	0 or 1	Selected time periods
p_MonatErster (monat)	days of months	Number of days from the first day of the calendar year until the first day of each month—for example, 1 for January, 32 for February, 60 for March, 91 for April.
p_Bevoelkerung	number of people	Total population of Switzerland
p_MonatKrisenbeginn	1–12	Selected start of month when a food supply crisis begins according to a model variant where the food supply crisis is simulated
p_AnteilBisKBeginn(ek)	0–1	Share of crops until a crisis begins: = ifthen(sum(kr, p_MonatErster(kr)) <= p_ErsterErntetag(ek), 0, sum(kr, p_MonatErster(kr)) > p_LetzterErntetag(ek), 1, (sum(kr, p_MonatErster(kr)) – p_ErsterErntetag(ek))/(p_LetzterErntetag(ek) – p_ErsterErntetag(ek) + 1))
p_KorrekturfaktorP(pp)	0–1	Correction factor for the assumption on the impact of scenarios over time. The parameter enables to model the implementation of a scenario over time. It includes values between 0 (in 2019) and 1 (in 2050)
Crops		
p_AktFlaeche(ek,inte,eg)	ha/production method/region	Current area of each crop by production methods and regions
p_FlaecheFix(ek)	0 or 1	Crops with a fixed area
p_FlaechenMaxima(p,eg,kg)	ha/region/year	Maximum crop group area by regions per year
p_Maximum_FF(kg)	%	Maximum area share of crop groups
p_Bedarf_SG(ek,sg)	kg/ha	Seed requirements for crops
p_isSaehPeriode(p,sg)	0 or 1/year	Crops that can be sown in a certain period
p_ErsterSaehtag(ek)	days (-183 to 183)	First sowing days of crops, where negative values refer to days in a previous year
p_ErstSaehCorr(ek)	days (1 to 366)	Recalculated first sowing day of crops: = p_ErsterSaehtag(ek) + (365 $ (p_ErsterSaehtag(ek)<0))
p_ErsterErntetag(ek)	days (1 to 366)	First harvest day of crops
p_LetzterErntetag(ek)	days (1 to 366)	Last harvest day of crops
p_Ertrag(ek,eg,prp)	ton/region/ha	Yield of raw products of crops

Parameter name	Unit	Description
p_PflanzErtrag(p,ek,eg,prp)	ton/region/ha/year	Crop yields per raw products, by regions and over time
p_AnteilErtrag(p,ek)	0–1/year	Percentage of annual yield that occurs in the period under consideration: = ifthen (ord(p) = 1, 1-p_AnteilBisKBeginn(ek), 1)
p_AktSaatgutBedarf(sg)	ton	Aggregation of seed requirements across crops with the same seed type: = 1 + sum((eg,ek), (p_Bedarf_SG(ek,sg) × p_AktFlaeche(ek, eg)/1000))
p_AnteilBisKBeginn(ek)	0-1	Percentage of the harvest period that precedes the onset of the scenario: = ifthen (sum(kr, p_MonatErster(kr)) <= p_ErsterErntetag(ek), 0, sum(kr, p_MonatErster(kr))>p_LetzterErntetag(ek), 1, (sum(kr, p_MonatErster(kr)) – p_ErsterErntetag(ek))/(p_LetzterErntetag(ek)–p_ErsterErntetag(ek) + 1))
p_WirdGesaeht(p,ek)	0 or 1/year	Crops sown in a certain period (if crops are sown in the same period as the seeds are produced, then transfer the value from set sg to ek))
p_ObsGemCon(ek)	ha	Maximum area of fruits, berries and vegetables
Livestock		
p_AktBestand(tk)	livestock	Current number of each livestock category
p_FaktorGVE(tk)	0–1	Factor of the number of livestock units
p_AktGVE(tk)	livestock	Current number in livestock units of each livestock category: = p_FaktorGVE (tk) × p_AktBestand (tk)
p_DauerTK(tk)	years	Lifespan of livestock
p_AnteilJahr(tk)	%	Duration belonging to the livestock category (if less than 1 year)
p_Reproduktionsrate(mt)	livestock	Offspring number per mother livestock and year
p_DauerLetzterWurfSchlachtung(mt)	days	Duration between last litter and slaughter
p_DauerZw2Wuerfen(mt)	days	Duration between two litters
p_MinReproRate(mt,nk)	%	Minimum share of offsprings
p_MaxReproRate(mt,nk)	0–1	Maximum proportion of offspring (default value is 1)
p_MinRemontRate(vg,fg)	0–1	Minimum share of the previous generation that becomes the next generation
p_MaxRemontRate(vg,fg)	0–1	Maximum share of the previous generation that becomes the next generation (default 1)
p_MinSchlachtRate(nt)	0–1	Minimum share of slaughter per livestock
p_MaxSchlachtRate(nt)	0–1	Maximum share of slaughter per livestock (default 1)

Parameter name	Unit	Description
p_TierErtrag(tk,trp)	kg/livestock	Raw product output of livestock
p_SchlachtErtrag(tk,trp)	kg/livestock	Raw product output of slaughtered livestock
p_KorrekturReproduktion(pp,mt)	0–1/year	Share of a period in which the mother livestock produces offspring on average: = ifthen(p_DauerLetzterWurfSchlachtung(mt) < p_Tage(pp), 1-p_DauerLetzterWurfSchlachtung(mt)/p_Tage(pp), 0)
p_cBestand(pp,tk)	0–1/year	Percentage of the young livestock at the end of the previous period staying in the current livestock category (calculated based on the duration of the growth phase of the livestock category and the duration of the period): = ifthen(p_Tage(pp) < 365 × p_AnteilJahr(tk), (p_AnteilJahr(tk) - p_Tage(pp)/365)/p_AnteilJahr(tk), 0)
p_cZuwachs(pp,tk)	livestock/year	Percentage of the increase in young livestock of the current period staying in the current livestock category (calculated based on the duration of the growth phase of the livestock category and the duration of the period): = ifthen(p_Tage(pp) < 365 × p_AnteilJahr(tk), 1, p_AnteilJahr(tk)/(p_Tage(pp)/365))
p_cAnzahlWuerfe(pp,mt)	livestock/year	Number of litter within the period: $ (p_KorrekturReproduktion(pp,mt)>0) = ifthen(p_Tage(pp)/p_DauerZw2Wuerfen(mt) < 1, 1, floor(p_Tage(pp)/p_DauerZw2Wuerfen(mt)))
p_cNachkommenProWurf(pp,mt)	livestock/year	Number of offspring per litter number: = p_Reproduktionsrate(mt)/(365/p_DauerZw2Wuerfen(mt))
p_AnteilSchlachtTiereKummuliert(pp,mt,w)	0–1/year	Proportion of slaughterings within the period up to litter: $((ord(w) <= p_cAnzahlWuerfe(pp,mt)) and (p_KorrekturReproduktion(pp, mt) > 0)) = ifthen(p_cAnzahlWuerfe(pp, mt) = 1, 1, (ord(w) × p_DauerZw2Wuerfen(mt) - p_DauerLetzterWurfSchlachtung(mt))/(p_Tage(pp) – p_DauerLetzterWurfSchlachtung(mt)))
p_cAnteilSchlachtTiere(pp,mt,w)	0–1/year	Proportion of slaughtering within the period per litter: $ ((ord(w) <= p_cAnzahlWuerfe(pp, mt)) and (p_KorrekturReproduktion(pp, mt)>0)) = ifthen(ord(w) = 1, p_cAnteilSchlachtTiereKummuliert(pp, mt, w), ifthen(ord(w) = p_cAnzahlWuerfe(pp, mt), 1–p_cAnteilSchlachtTiereKummuliert(pp, mt, w–1), p_cAnteilSchlachtTiereKummuliert(pp, mt, w) – p_cAnteilSchlachtTiereKummuliert(pp, mt, w–1)))
p_Anteil_TK(tk,monat)	0–1/month	Monthly livestock share in the total number of current livestock (default 1)
p_ParamMaxWurst(tk)	1	Controlling maximum number of calves reared for sausage production and suckling pigs
p_ParamMaxTier(tk)	1	Controlling maximum number of milked goat and sheep

Parameter name	Unit	Description
Fixed products		
p_Menge_FP(fp)	ton	Fixed supply of certain products
p_Anteil_FP(fp,monat)	0–1/month	Fixed share of monthly supply of certain products
p_MengeMT_FP(fp,monat)	ton/month	Monthly fixed supply of products: = p_Anteil_FP(fp,monat) × p_Menge_FP(fp)
p_Angebot_FP(p,fp)	ton/year	Fixed supply of annually available certain products: = sum(monat, p_MengeMT_FP(fp,monat))
Storage		
p_Bestand_PLP (plp)	ton/year	Storage of the compulsory stock
p_Anteil_LP(lp,monat)	0–5/year	Monthly storage percentages related to the maximum or reference value (default 1)
p_Menge_LP(lp)	ton	Maximum or reference quantity in the storage
p_MengeMT_LP(lp,monat)	ton/year	Monthly storage amount. If no scenario with monthly intervention is included: = p_Anteil_LP(lp,monat) × p_Menge_LP(lp)
p_Bestand_LP(lp)	ton	Storage in the initial period: = p_MengeMT_LP (lp, “Januar”)
p_EndLPBestand(lp)	ton	Storage at the end of the modelled period. If no scenario with monthly intervention is included: = p_MengeMT_LP (lp, “Januar”)
p_VorratMT_SG(sg,monat)	ton/year	Monthly stock of seeds
p_Vorrat_SG(sg)	ton	Stock of seeds
p_Anteil_SG(sg,monat)	0–1/year	Monthly seed stock percentages in relation to the maximum or reference value
p_Kapazitaet_IST(ist)	ton/year	Storage and processing capacity
Processing		
p_Koeffizient_VP(ap,vp)	ton raw products/ton processed products	Processing coefficient includes the quantity of raw products (ap) required to produce processed products (vp)
p_MinimalAnteil(ap,vp)	0–1	Minimum share of processed product in the raw product
p_MaximalAnteil(ap,vp)	0–1	Maximum share of processed product in the raw product (default 1)
p_Koeffizient_KP(ap,vp,kp)	ton processed products/ton by-products	Processing coefficient: quantity of processed products (vp) required to produce by-products (kp)
p_MaxDirektVerarbeitung_RP(rp)	0–1	Maximum ratio of raw products processed in the production
p_MinDirektVerfuegbar_FM(fm)	0–1	Minimum ratio of fodder used in the processing

Parameter name	Unit	Description
p_MaxDirektVerfuegbar_FM(fm)	0–1	Maximum ratio of fodder used in the processing (default 1)
p_MinDirektVerfuegbar_NM(nm)	0–1	Minimum ratio of food products used in the processing
p_MaxDirektVerfuegbar_NM(nm)	from 0 to a certain positive value	Maximum ratio of food products used in the processing (default 1)
p_VerarbeitungsKapazitaet(ap,vp)	ton/month	Processing capacity for some raw crop products
Trade		
p_Exporte_HP(hp,monat)	ton/month	Monthly export of products including raw, processed and by-products
p_Importe_HP(hp,monat)	ton/month	Monthly import of products including raw, processed and by-products
p_Produktexporte(p,hp)	ton/year	Export of products including raw, processed and by-products: = sum(monat, p_Exporte_HP(hp,monat))
p_Produktimporte(p,hp)	ton/year	Import of products including raw, processed and by-products: = sum (monat, p_ Importe _HP (hp, monat))
p_Exporte_SG(sg,monat)	ton/month	Monthly seed exports
p_Importe_SG(sg,monat)	ton/month	Monthly seed imports
p_Saatgutexporte(p,sg)	ton/year	Seed exports: = sum(monat, p_Exporte_SG (sg,monat))
p_Saatgutimporte(p,sg)	ton/year	Seed imports: = sum(monat, p_Importe_SG (sg,monat))
p_AnteilObsGem(hp)	from 0 to certain positive value	Shares of fruits and vegetables in the import
p_ExpSus (hp)	0 or 1	Indicating the export of sweets
Fodder		
p_TSGehalt(fm)	0–1	Dry matter content of the fodder
p_Maximum_FS(tk)	kg/livestock/day	Maximum fresh fodder intake
p_Maximum_TS(tk)	kg/livestock/day	Maximum dry matter of fodder intake
p_Gehalt_FW(fm,fw)	kg(MJ)/kgDM	Nutrient (energy) value content of fodder
p_Bedarf_FW(tk,fw)	gm(MJ)/livestock/day	Fodder demand of animals
p_MindestDeckung_FW(fw)	0–1	Minimum use of feed in the first period
p_Mindestanteil(tk,fmg)	0–1	Minimum ratio of dry matter consumption per feed group
p_AnteilSoFuet(monat)	0–1/month	Ratio of summer feeding in each month
p_AktBedarf_FW(tk,fw)	ton/livestock/1st year	Fodder demand for animals in the first period

Parameter name	Unit	Description
p_MinKonsGF(tk)	0–1	Minimal consumption of preserved basic feed
p_MaxFM(fm,tk)	0–1	Maximal consumption of fodder
p_KGF(fm)	0 or 1	List of preserved basic feed
p_Kraftfutter(pr)	0 or 1	Use of concentrated feed
p_TierWiederk(tk)	0 or 1	List of ruminants for which the concentrated feed is imported
p_FutVerl(fm)	0-1	Share of fodder loss
Food consumption		
p_AktuelleRation(nm)	gm/person/day	Current food ration before food losses
p_SumAktuelleRationGem	gm/person/day	Current total consumption of food products: = sum(nm, p_AktuelleRation(nm) × p_ConGem(nm))
p_AktuellerVerzehr(nm)	gm/person/day	Current food ration after food losses
p_NMG_Inhalt(nm,nmg)	gm/person/day	Content of food groups in food products
p_NMS_Inhalt(nm,nms)	gm/person/day	Micronutrients contents in food products
p_NM_Kaloriengehalt nm)	kcal/100 gm	Calorie content of food products
p_FaktorAbfall(nm)	0–1	Share of food waste due to disposal of garbage
p_FaktorHaushalt(nm)	0–1	Share of food waste due to disposal at the household consumption
p_AvFWProd(nm)	0–1	Share of avoidable food loss at processing
p_RationPlusTierfett(nm)	0–1	Rations for offal and animal fat
p_Relax1	0–1	Percentage of calorie targets missed
p_NMS_Kaloriengehalt(nms)	kcal/gm	Calorie content of nutrients
p_NMG_Inhalt(nm,nmg)	gm of food group/gm of food	Type of food products in food groups
p_NMGVzMin(nmg)	gm/person/day	Minimal consumption of food groups
p_NMGVzMax(nmg)	gm/person/day	Maximal consumption of food groups
p_NMGVzGsMin(nms)	gm/person/day	Minimal consumption of food groups according to the food pyramid
p_NMGVzGsMax(nms)	gm/person/day	Maximal consumption of food groups according to the food pyramid
p_NMSVzMin(nms)	gm/person/day	Minimal consumption of nutrients
p_NMSVzMax(nms)	gm/person/day	Maximal consumption of nutrients

Parameter name	Unit	Description
p_MRV_Inhalt(nm,nms)	gm of nutrient/gm of food	Content of nutrients in food products, such as fat, sugar, and salt
p_DRI_Inhalt(nm,nms)	gm of nutrient/gm of food	Content of nutrients in food products, such as fiber, protein, vitamins, potassium, calcium, magnesium, iron, and iodine
p_HENI(nm)	index/gm	Health Nutritional Index value of food products
p_MaxNMRation (nm)	gm/person/day	Maximum consumption of food products
p_ConGem(nm)	0 or 1	Indicating the consumption of vegetables
Environment		
p_Nabzug(tk)	kgN/livestock	Nitrogen accumulation by livestock less average losses
p_Pabzug(tk)	kgP/livestock	Phosphorous accumulation by livestock less average losses
p_NKgHa(ek,eg)	kgN/ha/region	Nitrogen requirement for crops
p_P2O5KgHa(ek,eg)	kgP/ha/region	Phosphorous requirement for crops
p_AnteilHofN(ek,eg)	0-1/ha/region	Share of nitrogen requirement in the form of livestock manure
p_AnteilHofP(ek,eg)	0-1/ha/region	Share of phosphorous requirement in the form of livestock manure
p_UW_EK(ek,inte,eg,uw)	environmental impact units/production method/region/ha	Environmental impacts of crop cultivation area
p_UW_PRP(ek,eg,prp,uw)	environmental impact units/region/kg	Environmental impacts of crop products
p_UW_TK(tk,uw)	environmental impact units/livestock	Environmental impacts of livestock category
p_UW_TRP(tk, uw)	environmental impact units/kg	Environmental impacts of livestock products
p_UW_REM(vg,fg,uw)	environmental impact units/kg	Environmental impacts of livestock replacing the livestock in the next generation
p_UW_VRB(ap,vp,uw)	environmental impact units/kg	Environmental impacts of processed products
p_UW_KPD(ap,vp,kp,uw)	environmental impact units/kg	Environmental impacts of by-products
p_UW_IMP(hp,uw)	environmental impact units/kg	Environmental impacts of imported products
p_UW_EXP(hp,uw)	environmental impact unit/kg	Environmental impacts of exported products
p_UW_HFD(uw)	environmental impact units/kg	Environmental impacts of farm manure (replacement of fertilizer by farm manure)
p_UW_HFD_R(uw)	environmental impact units/kg	Environmental effects of farm manure – negative parameter (replacement of fertilizer by farm manure)

Parameter name	Unit	Description
p_PSM_OFG(ek,inte)	Pesticide risk index/production method	Pesticide application risk indicator for surface water
p_PSM_NL(ek,inte)	Pesticide risk index/production method	Pesticide application risk indicator for semi-natural habitat
p_PSM_GW(ek,inte)	Pesticide risk index/production method	Pesticide application risk indicator for groundwater
p_OspNproTo(hp)	kgN/ton	Import of nitrogen in products
p_OspPproTo(hp)	kgP/ton	Import of phosphorous in products
p_OspDueN(ek,inte,eg)	kgN/ha	Mineral nitrogen fertilizer for crops
p_OspDueP(ek,inte,eg)	kgP/ha	Mineral phosphorous fertilizer for crops
p_OspFixN(ek,inte,eg)	kgN/ha	Biological nitrogen fixation
p_OspDepN(ek)	kgN/ha	Nitrogen deposit by crops
p_OspDepP(ek)	kgP/ha	Phosphorous deposit by crops
p_OspPflanzeN(nm)	kgN/ton	Nitrogen by plant-based food products
p_OspPflanzeP(nm)	kgP/ton	Phosphorous by plant-based food products
p_OspTierN(nm)	kgN/ton	Nitrogen by animal-based food products
p_OspTierP(nm)	kgP/ton	Phosphorous by animal-based food products
Income and labor		
p_ErzeugungEK(ek,inte,eg)	1,000 CHF/production method/region/ha	Revenue from crops
p_ErzeugungTK(tk)	1,000 CHF/livestock	Revenue from livestock
p_KostenVarEK(ek,inte,eg)	1,000 CHF/ production method/region/ha	Variable costs of crops
p_KostenVarTK(tk)	1,000 CHF/livestock	Variable costs of livestock
p_KostenFixEK(ek,inte,eg)	1,000 CHF/production method/region/ha	Fixed costs of crops
p_KostenFixTK(tk)	1,000 CHF/livestock	Fixed costs of livestock
p_DirzahlEK(ek,inte,eg)	1,000 CHF/production method/region/ha	Direct payments for crops

Parameter name	Unit	Description
p_DirzahlTK(tk)	1,000 CHF/livestock	Direct payments for livestock
p_DiskKostFixTK(p)	0–1/year	Discount of fixed costs for livestock whose number reduces in the model
p_TageFamilieEK(ek,inte,eg)	days/production method/region/ha	Family working days needed for crops
p_TageFamilieTK(tk)	days/ha/livestock	Family working days needed for livestock
p_LaborProd(p)	0.65–1/year	Change in labor productivity over years
p_Exportpreis(hp)	CHF/ton	Export price of commodities, including raw and processed products
p_Importpreis(hp)	CHF/ton	Import price of commodities, including raw and processed products
p_ImpTransKost(hp)	CHF/ton	Transportation costs of import

3.3 Variables

The variables of the model include positive and free variables. Table 4 presents the list of positive variables, and Table 5 lists the free variables. The model variables are included in the GAMS file *Variables.gms*. In this GAMS file, first the names of the variables are listed, and next to them their description in German is given, and then their names in English are given in parentheses. Figure 7 presents an example of positive variables included in GAMS. Similarly, are included free variables.

```
Positive variables
v_Flaechen(p,ek,inte,eg) Flaechen EK Ha (area for  crops)
v_FlaechenKG(p,eg,kg)    Flaechen KG Ha (area for crop groups)
v_Saatgutbedarf(p,sg)    Saatgutbedarf Tonne (seed requirements)
v_AngebotFM(p,fm)        Angebot an Futtermitteln Tonne (feed supply)
v_Futtermengen(p,fm)     Futtermengen Tonne (feed quantities)
v_Tierbestaende(pp,tk)   Tiere per Ende Periode Stück
*                        (livestock at the end of period)
;
```

Figure 7: Example of positive variables listed in Variables.gms.

Table 4: List of positive variables utilized in the model.

Positive variables	Units	Description
Crops		
v_Flaechen(p,ek,inte,eg)	ha/production method/region/year	Crop area
v_FlaechenKG(p,eg,kg)	ha/region/year	Crop group area
v_Erntemengen(p,ek,inte,eg,prp)	ton/production method/region/year	Crop harvest volumes
v_Bruttoerntemengen(p,inte,prp)	ton/production method/year	Gross crop amount
v_InlandPrdmengen(p,rp)	ton/year	Domestic production
v_Saatgutproduktion(p,sg)	ton/year	Quantity of crops used as seed
v_Saatgutbedarf(p,sg)	ton/year	Seed requirements for crops
v_FlaechenDev(p,ek,inte,eg)	ha/production method/region/year	Positive relative deviation between modelled and observed crop area
v_FlaechenDev1(p,ek,inte,eg)	ha/production method/region/year	Negative relative deviation between modelled and observed crop area
Livestock		
v_Tiere(pp,tk)	livestock/year	Livestock at the end of the period (including livestock for slaughter and replacement with the next generation)
v_Tierbestaende(pp,tk)	livestock/year	Livestock at the end of the period (not including livestock for slaughter and replacement with the next generation)
v_Durchschnittsbestaende(pp,tk)	livestock/year	Average livestock number
v_Tierproduktemengen(pp,tk,trp)	ton/livestock/year	Livestock products (milk, eggs) per livestock category and product
v_Remonten(pp,tk,tk)	livestock/year	Replacement of livestock with the next generation livestock
v_SchlachtTiere(pp,tk)	livestock/year	Slaughtered livestock

Positive variables	Units	Description
v_SchlachtTierePlus(pp,tk)	livestock/year	Livestock that are additionally slaughtered
v_Schlachtproduktemengen(pp,tk,trp)	ton/livestock/year	Products from slaughtered livestock
v_Nachkommenschaft(pp,mt)	livestock/year	Mother livestock category
v_NachkommenVonSchlachtTieren(pp,mt)	livestock/year	Mother livestock category that is left after slaughter
v_JungTiere(pp, nk)	livestock/year	Offspring livestock
v_JungeVonMuttertieren(pp,mt,nk)	livestock/livestock/year	Offspring livestock per mother livestock
v_TierbestPos(pp,tk)	livestock/year	Increase in the number of livestock with respect to the actual (input data) livestock number
v_TierbestNeg(pp,tk)	livestock/year	Decrease in number of livestock with respect to the actual (input data) livestock number
v_TierbestaendeDev(pp,tk)	livestock/year	Positive relative deviation between modeled and observed livestock number
v_TierbestaendeDev1(pp,tk)	livestock/year	Negative relative deviation between modeled and observed livestock number
Fodder		
v_FutterFuerTiere(p,fm,tk)	ton/livestock/year	Fodder for livestock
v_AngebotFW(p,tk,fw)	ton/livestock/year	Fodder nutrient supply for livestock
v_BedarfFW(p,tk,fw)	ton/livestock/year	Fodder nutrient requirement for livestock
v_AngebotTS(p,tk)	livestock/year	Livestock for whom fodder dry matter is supplied
v_AngebotTSjeFMG(p,tk,fmg)	ton/livestock/year	Supply of fodder dry matter by fodder group for livestock
v_AngebotFM(p,fm)	ton/year	Fodder supply from processed products and by-products
v_Futtermengen(p,fm)	ton/year	Fodder supply for livestock
v_Kraftfutterproduktion(p)	ktDM/year	Supply of concentrated feed
Storage		
v_Pflichtlagerbestaende(p,plp)	ton/year	Compulsory stocks at the end of each period
v_AusPflichtlager(p,plp)	ton/year	Withdrawal of products from the compulsory stock
v_Lagerbestaende(p, lp)	ton/year	Storage of products at the end of each period
v_InLager(p,lp)	ton/year	Placing commodities into storage
v_AusLager(p, lp)	ton/year	Withdrawal of products from storage
Processing		
v_Produktmengen(p,rp)	ton/year	Raw products available for processing
v_VerarbPrdMengen(pp,ap,vp)	ton/year	Processed product quantities per starting and processed product
v_NichtVerarbeitet(p,ap)	ton/year	Unprocessed products
v_KoppelPrdMengen(p,ap,vp,kp)	ton/year	By-products per raw and processed products

Positive variables	Units	Description
v_AngebotNM(p,nm)	ton/year	Food supply from processed products and by-products
v_VermVerlustProd(p,nm)	ton/year	Food waste during processing and by-product production stages
Trade		
v_ImporteTot(p,hp)	ton/year	Import quantity of products
v_Exporte(p,hp)	ton/year	Export quantity of products
v_importeTotAll(p)	ton/year	Import of all products
v_ImportFM(p)	ktDM/year	Import of fodder
v_ExporteDev(p,hp)	ton/year	Positive relative deviation between modelled and observed export
v_ExporteDev1(p,hp)	ton/year	Negative relative deviation between modelled and observed export
v_ImporteDev(p,hp)	ton/year	Positive relative deviation between modelled and observed import
v_ImporteDev1(p,hp)	ton/year	Negative relative deviation between modelled and observed import
Food supply and consumption	**ton/year**	
v_Nahrungsmengen(p,nm)	ton/year	Supply of food products
v_NMRationen(p,nm)	gm/person/day	Supply of food products for consumption
v_KalorienAngebot(p)	kcal/person/day	Calorie supply
v_NMGRationen(p,nmg)	gm/person/day	Consumption of food groups
v_NMSRationen(p,nms)	gm/person/day	Nutrient consumption
v_NMSKalorien(p,nms)	kcal/person/day	Nutrient consumption in calories
v_KalorienAngebotVz(p)	kcal/person/day	Calorie consumption
v_NMVerzehr(p,nm)	gm/person/day	Consumption of different food products
v_NMRationenGemSum(p)	gm/person/day	Total consumption of vegetables
v_DN(p,nm,nms)	gm/person/day	Consumption of disqualifying nutrients by food under maximum Reference values that contribute negatively to nutrient rich food index
v_LIM(p,nm,nms)	gm/person/day	Limiting nutrient index: consumption of disqualifying nutrients by food under maximum reference values that contribute negatively to nutrient rich food index
v_LIMsum(p,nms)	gm/person/day	Sum of limiting nutrient index: total food consumption of disqualifying nutrients under maximum reference values that contribute negatively to nutrient rich food index
v_QN(p,nm,nms)	gm/person/day	Consumption of qualifying nutrients by food under dietary reference intakes that contribute positively to nutrient rich food index
v_NR(p,nm,nms)	gm/person/day	Nutrient rich index: consumption of qualifying nutrients by food under dietary reference intakes that contribute positively to nutrient rich food index

Positive variables	Units	Description
v_NRsum(p,nms)	gm/person/day	Sum of nutrient rich index: total food consumption of qualifying nutrients under dietary reference intakes that contribute positively to nutrient rich food index
v_NMRationenDev(p, nm)	gm/person/day	Positive relative deviation between modeled and observed food consumption
v_NMRationenDev1(p,nm)	gm/person/day	Negative relative deviation between modeled and observed food consumption
Environment		
v_BedarfN(p)	kgN/year	Nitrogen use for crops
v_BedarfNHof(p)	kgN/year	Manure nitrogen use for crops
v_AnfallNHof(p)	kgN/year	Supply of nitrogen manure from livestock
v_BedarfP(p)	kgP/year	Phosphorous use for crops
v_BedarfPHof(p)	kgP/year	Manure phosphorous use for crops
v_AnfallPHof(p)	kgP/year	Supply of phosphorous manure from livestock
v_DifferenzNHofPos(p)	kgN/year	Excess supply of manure nitrogen in relation to crop demand
v_DifferenzNHofNeg(p)	kgN/year	Shortage supply of manure nitrogen in relation to crop demand
v_Umwelt_HFD(p,uw)	environmental impact units/year	Environmental effects of excess supply of manure nitrogen
v_Umwelt_HFD_N(p,uw)	environmental impact units/year	Environmental effects of shortage supply of manure nitrogen
v_PSM_OFG(p,ek,inte)	index/production method/year	Plant protection risk indicator for surface water
v_PSM_NL(p,ek,inte)	index/production method/year	Plant protection risk indicator for semi-natural habitat
v_PSM_GW(p,ek,inte)	index/production method/year	Plant protection risk indicator for groundwater
v_THGKonsum(p)	ktCO_2e/year	Total greenhouse gas (GHG) emissions from all stages of the food system
v_THG(p)	ktCO_2e/year	Agricultural GHG emissions
v_NH3(p)	ktNH_3/year	Agricultural ammonia emissions
v_NO3(p)	ktNO_3/year	Agricultural nitrate emissions
v_N2O(p)	ktN_2O/year	Agricultural nitrous oxide emissions
v_InertN(p)	ktN/year	Agricultural inert nitrogen emissions
v_OsparFuN(p)	tN/year	Nitrogen import from fodder
v_OsparFuP(p)	tN/year	Phosphorous import from fodder
v_OsparFixN(p)	tN/year	Fixed nitrogen from crops
v_OsparDepN(p)	tN/year	Nitrogen deposit in agriculture
v_OsparDepP(p)	tN/year	Phosphorous deposit in agriculture
v_OsparPflanzeN(p)	tN/year	Nitrogen removal from plants
v_OsparPflanzeP(p)	tN/year	Phosphorous removal from plants

Positive variables	Units	Description
v_OsparTierN(p)	tN/year	Nitrogen removal from livestock
v_OsparTierP(p)	tN/year	Phosphorous removal from livestock
v_OsparInputN(p)	tN/year	Total nitrogen input
v_OsparInputP(p)	tN/year	Total phosphorous input
v_OsparOutputN(p)	tN/year	Total nitrogen output
v_OsparOutputP (p)	tN/year	Total phosphorous output
v_OsparDueN (p)	tN/year	Nitrogen through mineral fertilizer
v_OsparDueP (p)	tN/year	Phosphorous through mineral fertilizer
Self-sufficiency		
v_Kalorienverbrauch(p)	mln. kcal/year	Calorie consumption
v_Kalorienimport(p)	mln. kcal/year	Calorie import
v_Kalorienexport(p)	mln. kcal/year	Calorie export
v_Kalorienproduktion(p)	mln. kcal/year	Domestic calorie production
Income, added value, and labor		
v_AktuelDirzahl	mln. CHF/year	Total current direct payments for agriculture
v_Familienarbeitstage(p)	1,000 days/year	Family working time used for agricultural production
v_ExporteHochpreis(p,hp)	1,000 CHF/ton	Export high cost assumptions for commodities
v_ExporteTiefpreis(p,hp)	1,000 CHF/ton	Export low cost assumptions for commodities
v_Exportreduktion(p,hp)	mln. CHF/ton	Export costs
v_ExporteTiefpreisAll(p)	1,000 CHF/ton	Export low cost assumptions for all commodities

Table 5: List of free variables used in the model.

Free variables	Units	Description
Objective function variables		
v_ZielMinRef	relative value	Reference variable aiming at minimization of deviations of selected variables from the observed situation (input data)
v_Umweltziel	environmental impact units	Minimization of environmental impacts
v_ZielLandEinkommen	mln. CHF	Maximization of agricultural domestic incomes
v_ZielAngebotNM	10,000 ton	Maximization of agricultural domestic production
v_ZielMinDev	relative value	Minimization of deviations of selected variables from the observed situation (input data)
Nutrient and health impacts		
v_NRF(p,nm,nms)	gm/person/day/year	Nutrient Rich Food Index 10.3
v_HENI(p,nm)	health index person/day/year	Health Nutritional Index by food products

Free variables	Units	Description
v_HENIsum(p)	health index person/day/year	Total Health Nutritional Index
Environment		
v_Umwelt_EK(p,uw)	environmental impact units/year	Environmental impacts of crop cultivation
v_Umwelt_PRP(p,uw)	environmental impact units/year	Environmental impacts of crop products
v_Umwelt_TK(p,uw)	environmental impact units/year	Environmental impacts of livestock number
v_Umwelt_TRP(p,uw)	environmental impact units/year	Environmental impacts of livestock products
v_Umwelt_REM(p,uw)	environmental impact units/year	Environmental impacts of products from replacement livestock
v_Umwelt_VRB(p,uw)	environmental impact units/year	Environmental impacts of processing
v_Umwelt_IMP(p,uw)	environmental impact units/year	Environmental impacts of import
v_Umwelt_EXP(p,uw)	environmental impact units /year	Environmental impacts of export
v_Umwelt_ZUB(p,uw)	environmental impact units/year	Environmental impacts of cooking
v_DifferenzPHofPos(p)	kgP/year	Excess supply of manure phosphorous in relation to crop demand

3.4 Equations

The list and definition of equations are included in the GAMS file *Equations.gms*. In this GAMS file, only the names of the equations are listed; next to them, their description in German are provided and then their names in English are given in parentheses. Figure 8 presents an example of the equations included in GAMS. We do not specify each equation here because a detailed description of equations is provided in section 4.

```
Equations
* ------------Pflanzenbau-Modell-Constraint---------------------------
e_pa(p,ek,inte,eg) Kulturen mit fixer Fläche Ha (crops with fixed area)
e_pb(p,eg,kg)      Aggregierung von Flaechen  ha (aggregation of crop areas)
e_pf_kw(p)         Flächenminima von Kunstwiesen (min area of artific meadows)
e_BioMax(p,ek)     Flächenmaxima Bio ha (maximum area of bio crops)
e_ph(p,ek,inte,eg,prp) Ernteergebnis (crop harvest amount)
;
```

Figure 8: Example of equations listed in Equations.gms.

4 Detailed description of the model equations

The model consists of different modules of the food system, including crop and livestock production, fodder for livestock, storage of products, trade of commodities, processing of raw products, environmental impacts, agricultural incomes, employment, domestic food self-sufficiency and food consumption. All these modules of the food system include specific constraint and balance equations. The equations are interlinked among the modules; moreover, the equations also include intertemporal relationships through the dynamics of variables.

The equations of the models are specified in the GAMS file named *Equations.gms*. The initial names of all equations start with *e_* notation and include the name of the equation—for example, *e_pa.* The name of equations can be written as preferred by the programmer.

4.1 Crop production

In the SWISSfoodSys model, the crop production area of Switzerland is divided into hill, valley, and mountain regions as well as according to production methods, such as conventional, extenso, and organic production. Crop yields differ depending on regions and production methods. Not all crops are suitable for all regions and not all of them can be grown under all production methods. Crop production is also classified into crop groups such as arable land, permanent crops, grains, potatoes, land suitable for meadow use, and others. Depending on the objective function, the cropping module optimizes the cropping pattern subject to available land area, three regions (i.e., hill, valley, and mountain), production methods (i.e., conventional, extenso, and organic), crop requirements such as seeds and fodder, processing, trade and food consumption. Each crop results in fixed yields over time because we do not include input-yield response functions. The model selects crops on a yearly basis, as long as the crop production constraints permit this.

A total of 43 different crops can be planted, including all major arable crops, such as wheat and feed grains, potatoes, sugar beet, rapeseed, silage corn, or vegetables. Further, crops are differentiated into 20 crop groups based on the crop types. Crop areas are restricted by the total available land use area, crop group areas, maximal and minimal permitted change in land use in each period, minimal and maximal crop areas, and area constraints for specific crops. Each equation of the crop module is described below.

Equation e_pa: Fixed area of crops

The area of certain permanent crops (specified by the parameter *p_FlaecheFix*) does not change over time. These crops include summer pastures in the Alps, scattered areas and other permanent crops. The area of the crops is equal to its actual area given in the parameter *p_AktFlaechen*. It remains constant over years, while the area of other crops can change.

```
e_pa(p,ek,inte,eg) $ (p_FlaecheFix(ek)<>0)..

         v_Flaechen(p,ek,inte,eg) $ ek_int_eg(ek,inte,eg)

         =E=

         p_AktFlaeche(ek,inte,eg) ;
```

Equation e_pb: Summation of areas belonging to the same crop group

All crops should correspond to the area of their respective crop group. Hence, the summed area of 43 crops by crop group is equal to the area of their corresponding individual crop groups. A crop can belong to more than one crop group.

```
e_pb(p,eg,kg)..

         sum((ek,inte), v_Flaechen(p,ek,inte,eg) $ EkZuKg(ek,kg) )

         =E=

         v_FlaechenKG(p,eg,kg) ;
```

Equation e_p1b: Maximum area of rotating crops

We assume there is a maximum share of specific arable crops or crop groups cultivated on the arable area in order to prevent pests and diseases. The area of these crop groups should be equal to or smaller than the maximum allowed share in the arable crop group area. The maximum shares are fixed over the periods.

```
e_p1b(p,eg,kg) $ (p_Maximum_FF(kg)<1)..

         v_FlaechenKG(p,eg,kg)

         =L=

         p_Maximum_FF(kg) * v_FlaechenKG(p,eg,"KG Ackerfläche") ;
```

Equation e_pc8b: Maximum increase in crop area in each period

Due to technical and organizational boundaries, we assume that the area of cultivation of the crops cannot drastically increase from one period to the next period. Therefore, we include an equation in which the area of crops in the current period should be equal to or lower than 130% of the area of crops in the previous period. Currently, this constraint is implemented only after the first period of the model—that is, 2019.

```
e_pc8b(p,ek,eg) $ (ord(p)>1)..

           sum(inte, v_Flaechen(p,ek,inte,eg) $ ek_int_eg(ek,inte,eg) )

         =L=

         1.3 * sum(inte, v_Flaechen(p-1,ek,inte,eg) $ ek_int_eg(ek,inte,eg) ) ;
```

Equation e_pc8c: Maximum decrease in crop area in each period

In this equation, we assume that the area of crop cultivation cannot drastically decrease from one period to the next period. The area of a crop in the current period should be equal or greater than 70% of the area of a crop in the previous period. This constraint is implemented only after the first period of the model—that is, 2019.

```
e_pc8c(p,ek,eg) $ (ord(p)>1)..

          sum(inte, v_Flaechen(p,ek,inte,eg) $ ek_int_eg(ek,inte,eg) )

         =G=

         0.7 * sum(inte, v_Flaechen(p-1,ek,inte,eg) $ ek_int_eg(ek,inte,eg) ) ;
```

Equation e_pd: Maximum crop group area

The increase in the area of certain crop groups is limited—for example, by the availability of land. These crop groups include total agricultural land, arable land, permanent crops, and crops grown in greenhouses. The upper boundaries can change over the periods—for example, due to the loss of agricultural land due to an increase of settlement areas.

```
e_pd(p,eg,kg) $ (p_FlaechenMaxima(p,eg,kg)>0)..

        v_FlaechenKG(p,eg,kg)

        =L=

        p_FlaechenMaxima(p,eg,kg) ;
```

Equation e_pf: Minimum crop cultivation area

We assume that currently grown crops should not disappear due to the optimization process of the model and certain areas of crops should remain. In the current version of the model, we assume that the required minimum cultivated area for crops should be greater or equal to the 40% area of the currently observed crop area. This constraint remains constant over periods.

```
e_pf(p,ek,inte,eg)..

        v_Flaechen(p,ek,inte,eg) $ ek_int_eg(ek,inte,eg)

        =G=

        0.4 * p_AktFlaeche(ek,inte,eg) $ ek_int_eg(ek,inte,eg)   ;
```

Equation e_pf1: Maximum crop cultivation area

To avoid certain crops occupying too large an area, we set a constraint where the maximum area of a crop can increase by threefold in the first period. This constraint is implemented in certain scenarios where drastic changes in the initial period are in place—for example, ban of imports of all food and feed products—in order to prevent unrealistic changes in land-use patterns.

```
e_pf1(p,ek,inte,eg) $ (ord(p)=1) ..

        v_Flaechen(p,ek,inte,eg) $ ek_int_eg(ek,inte,eg)

        =L=

        3 * p_AktFlaeche(ek,inte,eg) $ ek_int_eg(ek,inte,eg)   ;
```

Equation e_pf_kw: Minimum area of artificial meadow

Artificial meadow is important for livestock production and preservation of soil fertility, but it has a lower potential of calorie production and income generation than most other food crops. With this constraint, we restrict having a very low area or no area of artificial meadow in the model. We assume that the minimum percentage of artificial meadow (i.e., *Kunstwiesen*) should be greater or equal to a 22% area of currently observed arable crop group area (i.e., set element *KG Ackerfläche*). The value of 22% was defined as minimal share of artificial meadow in the crop rotation to prevent crops from pests and diseases as well as from yield losses (BRP and BLW, 1992).

```
e_pf_kw(p)..

        sum(eg, v_FlaechenKG(p,eg,"KG Kunstwiesen"))

        =G=

        0.22 * sum(eg, v_FlaechenKG(p,eg,"KG Ackerfläche"))  ;
```

We also use an alternative variant of minimum or stipulated percentage of the artificial meadow equation in scenario analysis, where the area of the artificial meadow is assumed to change over periods and is 30.5% of the area of observed arable crop group area in the initial model period but reduces to 22% in the last period. We assume that the share of artificial meadow area should be equal to the indicated share of the arable crop group. We also assume that the area of the artificial meadow reduces over years by using parameter *p_KorrekturfaktorP* (which includes values between 0 (in the initial period) and 1 (in that last period)). This equation can be relevant in scenarios of the analysis of the gradual impacts of policy implementation.

```
e_pf_kw(p)..

         sum(eg, v_FlaechenKG(p,eg,"KG Kunstwiesen"))

         =E=

         (0.305 - 0.085 * p_KorrekturfaktorP(p))
         * sum(eg, v_FlaechenKG(p,eg,"KG Ackerfläche")) ;
```

Equation e_ph: Crop output

Crop yields are based on the observed yields per hectare. Certain crops have no output (fallow land, hedges, ecological compensation areas). Crop yields differ by three regions (set *eg*, i.e., hills, valleys, and mountains) and production methods (set *inte*, i.e., conventional, extenso, and organic). Crop yields are fixed over years and do not change with respect to the input application levels (e.g., fertilizer) and other conditions (e.g., climate).

Depending on the scenarios and model objectives, it can be modelled that crop yields differ from the currently utilized yields or change over time. For this, crop yields (parameter *p_Ertrag*) are multiplied by the defined coefficients, which are represented as parameters. These coefficients can differ between crops and periods.

```
e_ph(p,ek,inte,eg,prp) $ ek_int_eg_prp(ek,inte,eg,prp)..

         p_Ertrag(ek,inte,eg,prp)
         * v_Flaechen(p,ek,inte,eg) $ ek_int_eg_prp(ek,inte,eg,prp)

         =E=

         v_Erntemengen(p,ek,inte,eg,prp) $ ek_int_eg_prp(ek,inte,eg,prp) ;
```

Equation e_pj: Total crop production

Gross crop harvest quantities are equal to the summed crop production, in tons (i.e., variable *v_Erntemengen*, where the harvest quantities in this variable are allocated to crops and regions with the same output type).

```
e_pj(p,inte,prp)..

         sum((ek,eg), v_Erntemengen(p,ek,inte,eg,prp)
         $ ek_int_eg_prp(ek,inte,eg,prp))

         =E=

         v_Bruttoerntemengen(p,inte,prp) ;
```

Equation e_pk: Allocation of raw crop products to seed or to food and feed products

Total crop production can be used as domestic crop product quantities consisting of food and feed products, and—optionally and only in the case of specific crops such as potatoes—as seed for crop cultivation. Here, the command *sameas* stipulates that the variable *v_Saatgutproduktion* includes those elements of set *sg* that also appear in set *prp*. The command *sameas* is similarly used in other equations of the model.

```
e_pk(p,prp)..

          sum(inte, v_Bruttoerntemengen(p,inte,prp) )

          =E=

          sum(sg, v_Saatgutproduktion(p,sg) $ sameas(sg,prp))

          + v_InlandPrdmengen(p,prp) ;
```

Equation e_pl: Crop seed requirement

Seed requirement for crops is calculated by considering seed requirements for each crop and crop-sown area. We include the seed requirement for 12 crops including winter wheat, summer wheat, rye, triticale, spelt, winter barley, oats, potatoes, field beans, protein peas, soy, rapeseed. Seed production is measured in tons, whereas seed requirement is in kg/ha; therefore, a division by 1,000 is needed to calculate seed requirements. Seed purchase has not yet been modelled.

```
e_pl(p,sg)..

           sum(ek, p_Bedarf_SG(ek,sg)
          * sum((inte,eg), v_Flaechen(p,ek,inte,eg)) $ ek_sg(ek,sg) ) / 1000

          =E=

          v_Saatgutproduktion(p,sg)  ;
```

Equation e_Pflanzkartof: Seed potato area

We assume that the total seed potato area cannot be larger than 30% of the total potato area because the model might select a larger area of seed potatoes than that is currently used for growing potato. This equation is formulated to prevent seed production that is not used. It is not implemented in the reference and certain other scenarios (for a description of objective functions, see section 5).

```
e_Pflanzkartof(p)..

          sum((inte,eg), v_Flaechen(p,"Pflanzkartoffeln",inte,eg) )

          =L=

          0.3 * sum((inte,eg), v_Flaechen(p,"Kartoffeln",inte,eg) ) ;
```

Equation e_PFlacZer: Crops should correspond to their regions and production methods

We assume that areas of crops should correspond to their initially defined regions and production methods (defined by the set *ek_int_eg*). Therefore, crops cannot grow in unsuitable regions and with unsuitable production methods.

```
e_PFlacZer(p,ek,inte,eg) $ (not ek_int_eg(ek,inte,eg))..

          v_Flaechen(p,ek,inte,eg)

          =E=

          0 ;
```

Equation e_pKGConsMax: Maximum crop group area

The maximum area of each crop group should not be 10% larger than the current total crop area. With this constraint, we restrict too large an increase in crop group areas.

```
e_pKGConsMax(p,eg,kg)..

          v_FlaechenKG(p,eg,kg)

          =L=

          1.1 * sum((ek,inte), p_AktFlaeche(ek,inte,eg))
          $ sum(ek, EkZuKg(ek,kg) ) ;
```

Equation e_ObstGemCon: Maximum area of fruits and vegetables

Fruits and vegetables are highly profitable and considered low GHG emission crops. In scenarios where agricultural incomes are maximized and/or GHG emissions need to reduce, the model might select large areas of fruits and vegetables, beginning from the initial period, which can be characterized by unrealistically high values. To address this issue, the maximum area of fruits and vegetables should be equal to or lower compared to their current observed area, which increases over periods and can be 200% larger in the last modelled period than in the initial period.

This constraint relaxes over time with the parameter *p_KorrekturfaktorP* (which has a value between 0 (in the initial period) and 1 (in the last period)). The constraint is relevant when analyzing scenarios in which the areas of fruits and vegetables become substantially larger than in the current situation.

```
e_ObstGemCon(p,ek)..

          p_ObsGemCon(ek) * sum((inte,eg), v_Flaechen(p,ek,inte,eg))

          =L=

          p_ObsGemCon(ek) * sum((inte,eg), p_AktFlaeche(ek,inte,eg))
          * (1 + 2 * p_KorrekturfaktorP(p)) ;
```

Equation e_FlaechTot: Total crop area should be equal to the total actual crop area

To have the same total area of modelled crops and actual crops, we include a constraint where the total crop area is equal to the total actual crop in each period of analysis.

```
e_FlaechTot(p)..

          sum((ek,inte,eg), v_Flaechen(p,ek,inte,eg))

          =E=

          sum((ek,inte,eg), p_AktFlaeche(ek,inte,eg)) ;
```

Equation e_BioMax: Maximum area of organic crops

The area of organic crops cannot increase by more than a certain percentage of the current area. This percentage increase is 0% in the initial period and increases to 100% in the last modelled period. We include this constraint because in certain scenarios where the environmental impacts of crops need to be reduced, the area of organic crops might substantially increase, thereby exceeding the demand for organic products, which is not included in the model. The relaxation of the constraint over time is formulated using the parameter *p_KorrekturfaktorP*, which has a value between 0 (in the initial period) and 1 (in the last period). This constraint is relevant for each crop that can be grown as organic.

```
e_BioMax(p,ek)..

         sum(eg, v_Flaechen(p,ek,"Bio",eg))

         =L=

         (1 + p_KorrekturfaktorP(p)) * sum(eg, p_AktFlaeche(ek,"Bio",eg)) ;
```

Equation e_WeizenMax: Maximum area of wheat

To avoid a large increase in wheat area (e.g., large increase in wheat area can be due to the fulfillment of the requirement of calorie consumption for people and environmental goals), we assume that the area of winter and summer wheats cannot increase by more than a certain percentage of the current area. This percentage is 5% in the initial period and increases to 205% in the last modelled period. The relaxation of the constraint over time is formulated using the parameter *p_KorrekturfaktorP*, which has a value between 0 (in the initial period) and 1 (in the last period).

```
e_WiezenMax(p)..

           sum((inte,eg), v_Flaechen(p,"Sommerweizen",inte,eg))
         + sum((inte,eg), v_Flaechen(p,"Winterweizen",inte,eg))

         =L=

         (1.05 + 2 * p_KorrekturfaktorP(p))
         * sum((inte,eg), p_AktFlaeche("Sommerweizen",inte,eg)
         + p_AktFlaeche("Winterweizen",inte,eg)) ;
```

Equation e_WeizenMin: Minimum area of wheat

Here, we set the lower boundary for wheat area. The minimum area of winter and summer wheats should not be lower than 95% of their actual area. This and the above mentioned constraints on winter and summer wheat areas are included to model the current assumption on the possible increase or decrease in their areas. These two equations are included depending on scenario settings.

```
e_WiezenMin(p)..

           sum((inte,eg), v_Flaechen(p,"Sommerweizen",inte,eg))
         + sum((inte,eg), v_Flaechen(p,"Winterweizen",inte,eg))

         =G=

         0.95
         * sum((inte,eg), p_AktFlaeche("Sommerweizen",inte,eg)
         + p_AktFlaeche("Winterweizen",inte,eg)) ;
```

Equation e_GerHafTriMax: Maximum area of winter barley, oat, and triticale

Similar to the constraint on wheat cultivation area given in equations *e_WeizenMax*, we assume that the area of winter barley, oat, and triticale cannot increase by more than a certain percentage of the current area. This percentage is 5% in the initial period and increases to 205% in the last modelled period. The relaxation of the constraint over time is formulated using the parameter *p_KorrekturfaktorP*, which has a value between 0 (in the initial period) and 1 (in the last period).

```
e_GerHafTriMax(p)..

          sum((inte,eg), v_Flaechen(p,"Wintergerste",inte,eg)
        + v_Flaechen(p,"Hafer",inte,eg)
        + v_Flaechen(p,"Triticale",inte,eg))

        =L=

        (1.05 + 2 * p_KorrekturfaktorP(p))
        * sum((inte,eg), p_AktFlaeche("Wintergerste",inte,eg)
        + p_AktFlaeche("Hafer",inte,eg)
        + p_AktFlaeche("Triticale",inte,eg)) ;
```

Equation e_GerHafTriMin: Minimum area of winter barley, oat, and triticale

The minimum area of winter barley, oat, and triticale cannot be lower than 95% of their current area. This and the above constraints on winter barley, oat, and triticale areas are included to reflect the current assumption on the possible increase or decrease in their areas. These two equations are included depending on the scenario settings of the model.

```
e_GerHafTriMin(p)..

          sum((inte,eg), v_Flaechen(p,"Wintergerste",inte,eg)
        + v_Flaechen(p,"Hafer",inte,eg)
        + v_Flaechen(p,"Triticale",inte,eg))

        =G=

        0.95
        * sum((inte,eg), p_AktFlaeche("Wintergerste",inte,eg)
        + p_AktFlaeche("Hafer",inte,eg)
        + p_AktFlaeche("Triticale",inte,eg)) ;
```

Equation e_HuelsenGemFlaec: Maximum area of legumes and vegetables

To limit the high increase in the area of legumes and vegetables due to their low current consumption and limited suitability for storage, respectively, the area of legumes and vegetables cannot increase by more than a certain percentage of the current area. This percentage is 0% in the initial period and increases to 200% in the last modelled period. The relaxation of the constraint over time is formulated using the parameter *p_KorrekturfaktorP*, which has a value between 0 (in the initial period) and 1 (in the last period). This constraint on legumes and vegetables areas reveals the current assumption regarding the possible increase in their areas. This equation is included depending on scenario settings.

```
e_HuelsenGemFlaec(p,ek,eg) $ (p_HuelsenGemFlaec(ek,eg)>0)..

        sum(inte, v_Flaechen(p,ek,inte,eg))

        =L=

        (1 + 2 * p_KorrekturfaktorP(p)) * sum(inte, p_AktFlaeche(ek,inte,eg)) ;
```

4.2 Livestock production

The livestock module includes the livestock production system. For livestock production, 41 livestock categories are distinguished. The rearing of young livestock to renew the herd is formulated in the model, as is the coverage of feed and nutrient requirements according to the livestock-specific feeding recommendations. Roughage and, thus, the yield of the meadow areas can be used by all categories of cattle (except fattening calves), horses, sheep, and goats. In contrast, pigs and poultry, rely on concentrated feed and by-products.

The livestock module includes the temporal relations between the production of livestock products, number of animals born, replacement rates, feed consumption, and slaughtering. Moreover, age, gender, breeds, and products produced are differentiated by livestock types. The relationships of the livestock module are at the monthly and daily levels (e.g., birth of animals). Figure 9 provides examples of the relationship among the age of cows, offspring, and next-generation cattle.

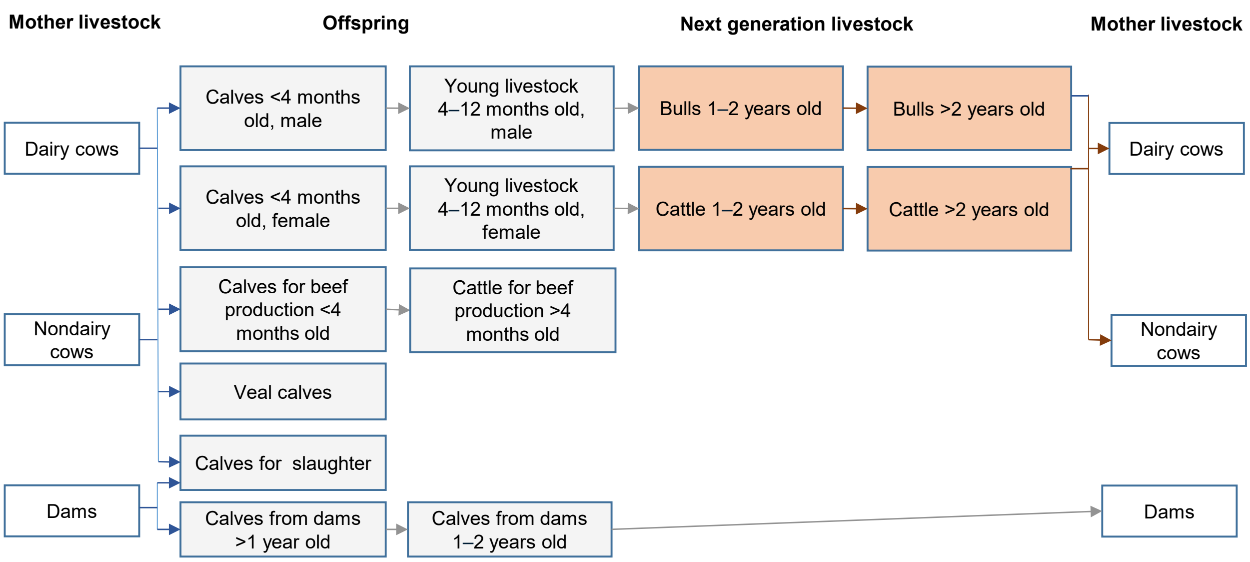


Figure 9: The relationship between cattle of different ages.

Equation e_ta: Livestock balance (A)

This constraint calculates the flows of different livestock variables that enter into a specific livestock category during the current period, without the subtraction of animals that leave the livestock category. These flows consider the livestock number at the end of the previous period (pp-1), young (newborn) livestock (i.e., *v_Jungtiere*), and replacement livestock with the next generation livestock (i.e., *v_Remonten*). In the first period of the model, the currently observed number of livestock is considered (i.e., *p_AktBestand*); in subsequent periods, the livestock number from the previous periods is included. Here, *v_Remonten* includes the sum of all previous generation animals that enter the livestock category.

```
e_ta(pp,tk)..

         p_AktBestand(tk) $ (ord(pp)=1)

         + v_Tierbestaende(pp-1,tk) $ (ord(pp)>1)

         + sum(nk, v_JungTiere(pp,nk) $ sameas(tk,nk) )

         + sum((vg,fg), ( (v_Remonten(pp,vg,tk) $ sameas(tk,fg)) $ tt(vg,fg)) )

         =E=

         v_Tiere(pp,tk) ;
```

Equation e_tb: Livestock balance (B)

Livestock number at the end of each period is calculated beginning from the variable *v_Tiere* and subtracting from it the slaughtered livestock and livestock that leaves the category to be included in the next-generation livestock category. Here, *v_Remonten* includes the sum of all next-generation animals that leave the livestock category.

```
e_tb(pp,tk)..

         v_Tiere(pp,tk)

         - v_SchlachtTiere(pp,tk)

         - sum((vg,fg), ( (v_Remonten(pp,tk,fg) $ sameas(tk,vg)) $ tt(vg,fg)) )

         =E=

         v_Tierbestaende(pp,tk) ;
```

Equation e_tc: Growth of livestock

The calculation of the growth of young livestock includes the change in different age livestock categories. Based on the livestock number at the end of the previous period, the livestock number at the end of the current period is determined in consideration of the animals that change from one livestock category to another during the period. Only a certain portion of the livestock number at the end of the previous period remains in that livestock category due to the growth of the animals (parameter *p_cBestand* includes a percentage of the livestock at the end of the previous period who remain in the current livestock category). In contrast, the number of animals in the livestock category increases with the livestock offspring or the livestock coming from the previous age category, respectively. However, only a certain portion of this increase remains in the livestock category if the duration of the livestock category is lower than the duration of the period (parameter *p_cZuwachs* includes a percentage of the increase in young livestock of the current period staying in the current livestock category).

```
e_tc(pp,tk) $ (not sum(nt, sameas(tk,nt)))..

       p_cBestand(pp,tk)

       * (p_AktBestand(tk) $ (ord(pp)=1)) $ (not sum(nt, sameas(tk,nt)) )

       + p_cBestand(pp,tk)

       * (v_Tierbestaende(pp-1,tk) $ (ord(pp)>1)) $ (not sum(nt, sameas(tk,nt)))

       + p_cZuwachs(pp,tk) * (sum(nk, v_JungTiere(pp,nk)
         $ sameas(tk,nk)) $ (not sum(nt, sameas(tk,nt)) )

       + sum((vg,fg), ( (v_Remonten(pp,vg,tk) $ sameas(tk,fg)) $ tt(vg,fg) ) )
       $ (not sum(nt, sameas(tk,nt)) )  )

       =E=

       v_Tierbestaende(pp,tk) $ (not sum(nt, sameas(tk,nt)) ) ;
```

Equation e_td: Minimum rate of livestock replacement by next generation

To avoid too small a number of livestock moving from one generation to the next generation, we include a minimum constraint on the replacement of livestock. The minimum ratio livestock moving to the next generation is a parameter and remains fixed over the modelled periods. This constraint is only used for less important livestock categories, such as rams.

```
e_td(pp,vg,fg) $ (tt(vg,fg) and p_MinRemontRate(vg,fg)>0)..

          p_MinRemontRate(vg,fg)
          * (v_Tiere(pp,vg) - v_Tierbestaende(pp,vg)) $ tt(vg,fg)

          =L=

          v_Remonten(pp,vg,fg) $ tt(vg,fg) ;
```

Equation e_te: Maximum rate of livestock replacement by the next generation

In this constraint, to have not too large a number of livestock moving from the previous generation to the subsequent generation, we restrict a maximum proportion of replacement livestock. This constraint is mainly used to ensure that not more than 50% of certain livestock categories move to subsequent female livestock categories.

```
e_te(pp,vg,fg) $ tt(vg,fg)..

          p_MaxRemontRate(vg,fg)
          * (v_Tiere(pp,vg) - v_Tierbestaende(pp,vg)) $ tt(vg,fg)

          =G=

          v_Remonten(pp,vg,fg) $ tt(vg,fg) ;
```

Equation e_tf: Maximum number of slaughtered livestock

The number of slaughtered livestock is limited—for example, due to limited slaughtering capacities. Therefore, the model has an upper limit of slaughtered livestock by ensuring a maximum proportion of livestock that is slaughtered (based on the livestock number at the end of the previous period and a useful life of a livestock).

```
e_tf(pp,nt)..

          p_Tage(pp) / 365 * p_MaxSchlachtRate(nt)
          * p_AktBestand(nt) $ (ord(pp)=1)

          + p_Tage(pp) / 365 * p_MaxSchlachtRate(nt)
          * v_Tierbestaende(pp-1,nt) $ (ord(pp)>1)

          =G=

          v_SchlachtTiere(pp,nt) ;
```

Equation e_tg: Minimum number of slaughtered livestock

Similar to constraint *e_tf*, here, a minimum proportion of slaughtered livestock are constrained based on the livestock number at the end of the previous period and the useful life of a livestock to avoid no slaughtering of livestock.

```
e_tg(pp,nt)..

          p_Tage(pp) / 365 * p_MinSchlachtRate(nt)
          * p_AktBestand(nt) $ (ord(pp)=1)

          + p_Tage(pp) / 365 * p_MinSchlachtRate(nt)
          * v_Tierbestaende(pp-1,nt) $ (ord(pp)>1)

          =L=

          v_SchlachtTiere(pp,nt) ;
```

Equation e_tii: Slaughtered livestock for offspring calculation

Equation *e_tii* calculates the auxiliary variable *v_SchlachtTierePlus*, which is used to calculate the number of offspring produced by dams (mother livestock). This auxiliary variable takes into account the duration between the last offspring and the slaughtered dam. During this time, there is no more offspring. Therefore, the auxiliary variable is lower than the actual number of slaughtered dams covered by the variable *v_SchlachtTiere*.

```
e_tii(pp,mt) $ (p_KorrekturReproduktion(pp,mt)>0)..

         v_SchlachtTierePlus(pp,mt)

         =E=

         p_KorrekturReproduktion(pp,mt) * v_SchlachtTiere(pp,mt) ;
```

Equation e_tij: Offspring from slaughtered dams

This equation calculates the number of offspring produced by dams who are subsequently slaughtered (mother livestock). The calculation takes into account the number of litters per year and considers specific shares of slaughtered livestock for each litter, including the nonconstant length of the time period, *pp*.

```
e_tij(pp,mt) $ (p_KorrekturReproduktion(pp,mt)>0)..

         sum(w $ (ord(w) <= p_cAnzahlWuerfe(pp,mt)),

         ord(w) * p_cNachkommenProWurf(pp,mt)
         * p_cAnteilSchlachtTiere(pp,mt,w)
         * v_SchlachtTierePlus(pp,mt) )

         =E=

         v_NachkommenVonSchlachtTieren(pp,mt)  ;
```

Equation e_tj: Number of offspring

The number of offspring can change in each period as a result of the reproduction rate of dams, slaughtered dams that produced offspring, and the difference between slaughtered dams and the average herd size of dams. The calculation of offspring per dam category is based on the average livestock population.

However, the number of offspring can also be calculated in an easier but still correct manner by adjusting the parameter *p_Reproduktionsrate*, where there is a certain time period between the last litter and the slaughtering of the dam. In this case, the parameter does not include the number of offspring per dam and year for the productive time span but the number of total offspring during the lifetime of the dam divided by the time span (number of years) from the moment the animal enters the dam livestock category until the slaughtering of the dam. This leads to a slightly lower parameter *p_Reproduktionsrate*. In this case, the variables *v_SchlachtTiere* and *v_NachkommenVonSchlachtTieren* does not have to be included in the equation *e_tj* and the variable *v_Tiere* has to be replaced by the variable *v_Durchschnittsbestaende*. Furthermore, the equations *e_tii* and *e_tij* can be omitted, as can several sets (*w*), parameters (*p_KorrekturReproduktion, p_cAnzahlWuerfe, p_cNachkommenProWurf, p_AnteilSchlachtTiereKummuliert, p_cAnteilSchlachtTiere), and variables (v_SchlachtTierePlus, v_Nachkommen VonSchlachtTieren*).

```
e_tj(pp,mt)..
          p_Tage(pp)/365 * p_Reproduktionsrate(mt)
          * (v_Tiere(pp,mt) - v_SchlachtTiere(pp,mt))
          + v_NachkommenVonSchlachtTieren(pp,mt)
          =E=
          v_Nachkommenschaft(pp,mt) ;
```

Equation e_tk: Total offspring per dam category

A dam in one category (mother livestock) can have offspring categorized in different offspring categories. This constraint summarizes the number of offspring categories to obtain the total number of offspring per dam category (*v_Nachkommenschaft*)—that is, the total number of offspring per dam category is allocated to different offspring categories.

```
e_tk(pp,mt)..
          v_Nachkommenschaft(pp,mt)
          =E=
          sum(nk, v_JungeVonMuttertieren(pp,mt,nk) $ rr(mt,nk) ) ;
```

Equation e_tl: Total offspring per offspring category

Different dams (mother livestock) may have offspring that belong to the same offspring category. This constraint summarizes the number of offspring of the dams to obtain the total number of offspring per offspring category (*v_JungTiere*).

```
e_tl(pp,nk)..
          v_JungTiere(pp,nk)
          =E=
          sum(mt, v_JungeVonMuttertieren(pp,mt,nk) $ rr(mt,nk) ) ;
```

Equation e_tm: Minimum reproduction rate

Dams can produce more than one type of offspring. For example, dairy cows can produce (1) rearing female calves under four months old, (2) rearing male calves under four months old, (3) calves for fattening under four months old, (4) veal calves, and (5) calves slaughtered for sausages. To have a certain share of offspring from dams, we impose a constraint of a minimum reproduction rate on dams that can produce more than one type of offspring. In the current version of the model, the minimum birth rate of the offspring of dams is assumed to be 0 for all possible combinations of dams and offspring.

```
e_tm(pp,mt,nk) $ (p_MinReproRate(mt,nk)>0)..
          p_MinReproRate(mt,nk) * v_Nachkommenschaft(pp,mt) $ rr(mt,nk)
          =L=
          v_JungeVonMuttertieren(pp,mt,nk) $ rr(mt,nk) ;
```

Equation e_tn: Maximum reproduction rate

Similar to the minimum reproduction rate, this constraint limits the upper share of offspring categories per dam. It is mainly used to limit the share of female offspring per dam to 50%.

```
e_tn(pp,mt,nk) $ (p_MaxReproRate(mt,nk)<1)..

         p_MaxReproRate(mt,nk) * v_Nachkommenschaft(pp,mt) $ rr(mt,nk)

         =G=

         v_JungeVonMuttertieren(pp,mt,nk) $ rr(mt,nk) ;
```

Equation e_tp: Average number of livestock

The number of livestock might change within the same period. We calculate the average number of livestock in each period as the average of the livestock number in the first period based on the actual livestock number; in subsequent periods, this is based on livestock number at the end of the previous period and at the end of the current period.

```
e_tp(p,tk)..

         0.5 * (

         p_AktBestand(tk) $ (ord(p)=1)

         + v_Tierbestaende(p-1,tk) $ (ord(p)>1)

         + v_Tierbestaende(p,tk) )

         $ sum(fw, tk_fw(tk,fw))

         =E=

         v_Durchschnittsbestaende(p,tk) $ sum(fw, tk_fw(tk,fw)) ;
```

Equation e_tq: Raw products from slaughtered livestock

Slaughtered livestock produces meat, bones, fat, and offal. The number of slaughtered livestock (regular and premature slaughtering) multiplied by the yield per slaughtered animal yields the production from slaughtered livestock. The parameter with information on yield from slaughtered livestock includes the amount of meat, bones, fat, and offal per slaughtered livestock measured in sales weight (kg per livestock number). Processing losses are already deducted from the yield of products of slaughtered livestock. Since the total production is measured in tons, a division by 1,000 is included.

```
e_tq(p,tk,trp) $ tk_sp(tk,trp)..

        (p_SchlachtErtrag(tk,trp) * v_SchlachtTiere(p,tk) $ tk_sp(tk,trp))/1000

        =E=

        v_Schlachtproduktemengen(p,tk,trp) $ tk_sp(tk,trp) ;
```

Equation e_tr: Raw products from live animals

Livestock also produces products without being slaughtered, such as milk and eggs. The quantities of these raw products from livestock in each period are calculated for the average size of livestock. The parameter *p_DauerP* includes the duration of the periods in years. A factor 1,000 is needed for the conversion of kg to tons.

```
e_tr(p,tk,trp)..

          ( p_DauerP(p) * p_TierErtrag(tk,trp)
          * (v_Durchschnittsbestaende(p,tk) $ sum(fw, tk_fw(tk,fw)))
          $ tk_tp(tk,trp) ) / 1000

          =E=

          v_Tierproduktemengen(p,tk,trp) $ tk_tp(tk,trp) ;
```

Equation e_ts: Domestic production of products from slaughtered livestock

Products from slaughtered livestock (e.g., meat, bones, fat, and offal) are summed per raw product over the livestock categories and considered as domestically produced products.

```
e_ts(p,trp)..

          sum(tk, v_Schlachtproduktemengen(p,tk,trp) $ tk_sp(tk,trp))

          =E=

          sum(rp, v_InlandPrdmengen(p,rp) $ sameas(trp,rp))
          $ sum(tk, tk_sp(tk,trp)) ;
```

Equation e_tt: Domestic production from live animals

Raw products from livestock such as milk and eggs are summed per raw product across livestock categories and considered as domestically produced agricultural products. The aggregated variable *v_InlandPrdmengen* includes the products from both the slaughtered and live animals.

```
e_tt(p,trp)..

          sum(tk, v_Tierproduktemengen(p,tk,trp) $ tk_tp(tk,trp) )

          =E=

          sum(rp, v_InlandPrdmengen(p,rp) $ sameas(rp,trp))
          $ sum(tk, tk_tp(tk,trp)) ;
```

Equation e_Tier_Min_1: Minimum number of livestock

In this constraint, we assume a minimum number of livestock to avoid the model not selecting certain livestock categories during the optimization process. The model assumes that the minimum number of livestock cannot be smaller than 20% of the currently observed livestock number. This percentage of the livestock number can change depending on the model scenarios and assumptions.

```
e_Tier_Min(pp,tk)..

          v_Tierbestaende(pp,tk)

          =G=

          0.2 * p_AktBestand(tk) ;
```

Equation e_Tier_MaxAll: Maximum number of livestock

The number of livestock might increase by excessively large values because rearing certain livestock categories might be more profitable and environmentally friendly and yield higher calorie and nutrient content than other livestock categories and crops. We consider the maximum number of livestock, which cannot be larger than 110% of currently observed livestock number, except for foals. This constraint is mainly utilized in scenarios where livestock number is constrained or expected to be reduced. The maximum percentage of the livestock number can be adjusted depending on the model scenarios and assumptions.

```
e_Tier_MaxAll(pp,tk) $ ( (p_AktBestand(tk)>0)
and (not sameas(tk,"Fohlen bei Fuss")) )..

         v_Tierbestaende(pp,tk)

         =L=

         1.1 * p_AktBestand(tk)  ;
```

Equation e_Tier_Min_3: Minimum population of young cattle

Young cattle are needed as a replacement of livestock in subsequent generations. To ensure a ratio between young cattle and dairy cows that is close to the current ratio, it is assumed that the oldest young cattle category cannot be lower than 15% of the dairy cow number (the statistical ratio is approximately 18%). This constraint still allows a decrease in the number of dairy cows, as it is possible to produce a beef or veal from young cattle instead of breeding dairy cows.

```
e_Tier_Min_3(pp)..

         v_Tierbestaende(pp,"Rinder über 2-jährig")

         =G=

         (
           v_Tierbestaende(pp,"Kühe zur Verkehrsmilchproduktion")
         + v_Tierbestaende(pp,"Kühe, gemolken, keine Verkehrsmilchproduktion")
         + v_Tierbestaende(pp,"Kühe int zur Verkehrsmilchproduktion")
         + v_Tierbestaende(pp,"Kühe ext zur Verkehrsmilchproduktion")
         ) * 0.15 ;
```

Equation e_Tier_MaxWurst: Maximum number of sausage calves and suckling pigs

After period (year) one, the stock of sausage calves and suckling pigs (these are the elements of *tk* for which the parameter *p_ParamMaxWurst>0*) should be 0. These two livestock categories enable the model to reduce livestock number fast in crisis scenarios such as in scenarios without import of food and feed products. They are not relevant in normal times without crisis scenarios.

```
e_Tier_MaxWurst(pp,tk) $ (ord(pp)>1)..

         p_ParamMaxWurst(tk) * v_Tierbestaende(pp,tk)

         =L=

         0 ;
```

Equation e_Tier_Max: Maximum number of milked goats and sheep

To avoid a large number of goats and sheep used for milking, we assume their stock number cannot be larger than 300% of their currently observed number.

```
e_Tier_Max(pp,tk)..

          p_ParamMaxTier(tk) * v_Tierbestaende(pp,tk)

          =L=

          3 * p_ParamMaxTier(tk) * p_AktBestand(tk) ;
```

Equation e_Tier_Max: Maximum amount of cow milk production

We assume that the maximum amount of modelled cow production should not be larger than the current amount. This constraint is used because there are cow varieties with different milk yields, and cow milk production is a highly profitable activity for farmers; this can lead to a high production amount of cow milk and a model selection of cows that cannot be reared in large number.

```
e_MaxMilch(p)..

          ( p_TierErtrag("Kühe zur Verkehrsmilchproduktion","Milch (RP)")
          * p_AktBestand("Kühe zur Verkehrsmilchproduktion")

          + p_TierErtrag("Kühe int zur Verkehrsmilchproduktion","Milch (RP)")
          * p_AktBestand("Kühe int zur Verkehrsmilchproduktion")

          + p_TierErtrag("Kühe ext zur Verkehrsmilchproduktion","Milch (RP)")
          * p_AktBestand("Kühe ext zur Verkehrsmilchproduktion")
          ) / 1000

          =G=

          ( p_TierErtrag("Kühe zur Verkehrsmilchproduktion","Milch (RP)")
          * v_Durchschnittsbestaende(p,"Kühe zur Verkehrsmilchproduktion")

          + p_TierErtrag("Kühe int zur Verkehrsmilchproduktion","Milch (RP)")
          * v_Durchschnittsbestaende(p,"Kühe int zur Verkehrsmilchproduktion")

          + p_TierErtrag("Kühe ext zur Verkehrsmilchproduktion","Milch (RP)")
          * v_Durchschnittsbestaende(p,"Kühe ext zur Verkehrsmilchproduktion")
          ) / 1000 ;
```

4.3 Feed

The feed module includes the relationships related to the feed consumed by livestock. Each livestock has a minimum level of energy and nutrient requirement. In addition, the minimum and maximum shares of specific feed or feed groups in the feed ration have to be met. The production quantity of livestock products and livestock weight do not change with the feed consumed—that is, the model does not have the feed–livestock response functions.

Equation e_fa: Total feed consumed

Total feed consumed is based on the sum of feed consumed by different livestock types. The feed types consumed by livestock is also characterized by losses (*p_FutVerl*) because not all produced fodder is eaten by livestock and some quantity of it remains.

```
e_fa(p,fm)..

          p_FutVerl(fm) * v_Futtermengen(p,fm) $ sum((tk,fmg), ft(fm,tk,fmg) )

          =E=

          sum(tk, v_FutterFuerTiere(p,fm,tk) $ sum(fmg, ft(fm,tk,fmg)) ) ;
```

Equation e_fb: Feed nutrient supply

The calculation of the feed nutrient supply per livestock category is based on the dry matter content of feed (*p_TSGehalt*), as the data on energy and nutrient content of the feed (*p_Gehalt_FW*) is related to the dry matter of the feed.

```
e_fb(p,tk,fw) $ tk_fw(tk,fw)..

         sum(fm, p_TSGehalt(fm) * p_Gehalt_FW(fm,fw)
         * v_FutterFuerTiere(p,fm,tk) $ sum(fmg, ft(fm,tk,fmg)) )
         $ tk_fw(tk,fw)

         =E=

         v_AngebotFW(p,tk,fw) $ tk_fw(tk,fw) ;
```

Equation e_fc: Feed energy and nutrient requirement

We calculate the total feed energy and nutrient requirements for livestock using the average livestock number and feed energy and nutrient requirement for each livestock type. The unit of the energy and nutrient requirement data (*p_Bedarf_FW*) is in MJ/animal/day (energy) and g/animal/day (nutrients), respectively. By means of the factors 365/*p_DauerP* and 1,000,000, the unit of energy and nutrient requirement is converted to TJ/year and tons/year, respectively. The energy unit is not the same for all livestock categories, as the available energy in the same feed product can be different for cows and pigs, for example. Therefore, set fw contains different energy units according to the units utilized for the livestock categories in the model.

```
e_fc(p,tk,fw) $ tk_fw(tk,fw)..

         ( 365 / p_DauerP(p) * p_Bedarf_FW(tk,fw)
         * v_Durchschnittsbestaende(p,tk) $ tk_fw(tk,fw) ) / 1000000

         =E=

         v_BedarfFW(p,tk,fw) $ tk_fw(tk,fw) ;
```

Equation e_fd: Feed balance

To avoid the overconsumption of feed by livestock, the livestock feed nutrient requirement should not be larger than the sum of feed nutrient supply for livestock. In addition, the energy and nutrient supply per livestock category must be equal or higher than the corresponding requirement. In certain scenarios (e.g., crisis scenarios), an undersupply of less important nutrient may be tolerated (temporarily) in the model. For such scenarios, the equation (or the data on nutrient requirement) may need to be adjusted accordingly.

```
e_fd(p,tk,fw) $ tk_fw(tk,fw)..

         v_AngebotFW(p,tk,fw) $ tk_fw(tk,fw)

         =G=

         v_BedarfFW(p,tk,fw) $ tk_fw(tk,fw) ;
```

Equation e_ff: Maximum intake of feed fresh matter

For certain livestock categories (e.g., pigs), the feeding restrictions include a maximum feed intake per livestock and day measured in kg of fresh matter. For these livestock categories, the feed supply must be equal or lower than the maximum fresh matter intake for the livestock category (measured in tons). The parameter for maximum feed fresh matter intake by livestock categories (*p_Maximum_FS*) is fixed and does not change over periods.

```
e_ff(p,tk) $ (p_Maximum_FS(tk)>0)..

          sum(fm, v_FutterFuerTiere(p,fm,tk) $ sum(fmg, ft(fm,tk,fmg)) )

          =L=

          (365 / p_DauerP(p) * p_Maximum_FS(tk) * v_Durchschnittsbestaende(p,tk)
          $ sum(fw, tk_fw(tk,fw)) / 1000) $ (sum((fm,fmg), ft(fm,tk,fmg)) ) ;
```

Equation e_fg: Supply of total feed dry matter by livestock and feed categories

This constraint calculates the supply of total feed dry matter to livestock differentiated by livestock categories and feed groups.

```
e_fg(p,tk,fmg)..

          sum(fm, p_TSGehalt(fm) * v_FutterFuerTiere(p,fm,tk) $ ft(fm,tk,fmg))

          =E=

          v_AngebotTSjeFMG(p,tk,fmg) $ sum(fm, ft(fm,tk,fmg)) ;
```

Equation e_fh: Supply of total feed dry matter by livestock category

This constraint calculates the total feed dry matter supply differentiated by livestock category.

```
e_fh(p,tk)..

          sum(fmg, v_AngebotTSjeFMG(p,tk,fmg) $ sum(fm, ft(fm,tk,fmg)) )

          =E=

          v_AngebotTS(p,tk) $ sum((fm,fmg), ft(fm,tk,fmg)) ;
```

Equation e_fi: Maximum intake of feed dry matter

For most of the ruminant livestock categories, the feeding restrictions include a maximum feed intake per animal and day measured in kg of dry matter. For these livestock categories, the supply of dry matter feed should not be larger than the potential maximum intake (measured in tons of dry matter). The parameter on the maximum dry matter intake by livestock categories (*p_Maximum_TS*) is fixed and does not change over periods.

```
e_fi(p,tk) $ (p_Maximum_TS(tk)>0)..

          v_AngebotTS(p,tk) $ sum((fm,fmg), ft(fm,tk,fmg))

          =L=

          (365 / p_DauerP(p) * p_Maximum_TS(tk) * v_Durchschnittsbestaende(p,tk)
          $ (sum(fw, tk_fw(tk,fw)) and sum((fm,fmg), ft(fm,tk,fmg)) ) ) / 1000 ;
```

Equation e_fk: Minimum share of feed groups in feed ration

In order to ensure realistic feed rations, specific shares of feed groups in the feed ration have to be met in addition to the energy and nutrient requirements and the maximum intake constraints. Moreover, a minimum share of winter feeding has to be met by the ruminants. Therefore, the model assumes that at least a specific share (*p_MinKonsGF*) of preserved feed dry matter such as dried feed for balanced mixed stock, extensively produced dried feed, grass silage for balanced mixed stock, and dough ripe maize silage (*p_KGF*>0 for these feed products) must be fed to ruminants.

This equation could be applied to additional feed groups that are defined by the set *fmg*. To do so, the first part of the equation should be replaced by the variable *v_AngebotTSjeFMG*, which includes the feed quantity per feed group. The parameter in the second part of the equation (*p_MinKonsGF*) should be replaced by the parameter *p_Mindestanteil*, which includes the minimum share of the corresponding feed group.

```
e_fk(p,tk) $ (p_MinKonsGF(tk)>0)..

         sum(fm, p_TSGehalt(fm) * p_KGF(fm) * v_FutterFuerTiere(p,fm,tk)
         $ sum(fmg, ft(fm,tk,fmg)) )

         =G=

         p_MinKonsGF(tk) * v_AngebotTS(p,tk) ;
```

Equation e_fl: Maximum share of single feed products in feeding ration

We assume that each livestock category can be fed with a maximum fodder dry matter, which should not be larger than a certain share of supply of fodder dry matter. With this constraint, we restrict overfeeding the livestock by specific fodder type.

```
e_fl(p,fm,tk) $ (p_MaxFM(fm,tk)>0)..

         p_TSGehalt(fm) * v_FutterFuerTiere(p,fm,tk) $ sum(fmg, ft(fm,tk,fmg))

         =L=

         p_MaxFM(fm,tk) * v_AngebotTS(p,tk) $ sum(fmg, ft(fm,tk,fmg)) ;
```

Equation e_MinKF_KuhInt: Minimum concentrate fodder fed to cows with high milk yield

A high quantity of milk producing cows is usually fed with concentrate fodder. Hence, this livestock category needs to be fed with a minimum quantity of concentrate fodder, which should have a larger amount than 10% of fodder dry matterthat is fed to livestock. We use this constraint in scenarios without imports of concentrate fodder to ensure the feeding of concentrate fodder to cows with a high milk yield.

```
e_MinKF_KuhInt(p)..

         sum(fm, p_Kraftfutter(fm) * p_TSGehalt(fm)
         * v_FutterFuerTiere(p,fm,"Kühe int zur Verkehrsmilchproduktion")
         $ sum(fmg, ft(fm,"Kühe int zur Verkehrsmilchproduktion",fmg)) )

         =G=

         0.1 * sum(fm, p_TSGehalt(fm)
         * v_FutterFuerTiere(p,fm,"Kühe int zur Verkehrsmilchproduktion")
         $ sum(fmg, ft(fm,"Kühe int zur Verkehrsmilchproduktion",fmg)) ) ;
```

Equation e_MinKF_KuhVMP: Minimum concentrate fodder fed to cows with average milk yield

Milk-producing cows that produce an average quantity of milk are also fed with concentrate fodder. Accordingly, we assume a constraint, where milk-producing cows with average milk yield need to be fed with a minimum quantity of concentrate fodder, which should have an amount larger than 5% of fodder dry matter fed to livestock. This constraint is used in scenarios where there are no imports of concentrate fodder to ensure feeding of concentrate fodder to milk-producing cows with an average milk yield.

```
e_MinKF_KuhVMP(p)..

          sum(fm, p_Kraftfutter(fm) * p_TSGehalt(fm)
          * v_FutterFuerTiere(p,fm,"Kühe zur Verkehrsmilchproduktion")
          $ sum(fmg, ft(fm,"Kühe zur Verkehrsmilchproduktion",fmg)) )

          =G=

          0.05 * sum(fm, p_TSGehalt(fm)
          * v_FutterFuerTiere(p,fm,"Kühe zur Verkehrsmilchproduktion")
          $ sum(fmg, ft(fm,"Kühe zur Verkehrsmilchproduktion",fmg)) ) ;
```

Equation e_MaxKF_Wiederk: Maximum concentrate fodder fed to ruminants

Here, we assume that the maximum fodder dry matter concentrate can be fed to ruminants, which enables the avoidance of excessively large values of fodder concentrate given to ruminants. This equation includes concentrate fodder that is also not produced as a by-product. The constraint assumes that the concentrate fodder given to ruminants should be less than the amount of oil and sugar beet molasses fodders as well as less than the twice the amount of grain fodder given to ruminants. Such conditions might be implemented over time and, thus, the constraint is implemented over modelled periods using parameter *p_KorrekturfaktorP*.

This constraint is modelled when analyzing the restriction of the import of concentrate fodder import in situations where the system does not include the import of concentrate fodder. The constraint can be switched on/off depending on the scenario settings.

```
e_MaxKF_Wiederk(p,tk)..

          sum(fm, p_Kraftfutter(fm) * p_TSGehalt(fm) * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,fm,tk) $ (sum(bf, not sameas(fm,bf)) )  )

          =L=

          2 * sum(mnp, p_TSGehalt(mnp) * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,mnp,tk) ) * (1+9*(1-p_KorrekturfaktorP(p)) )

          + sum(oel, p_TSGehalt(oel) * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,oel,tk) ) * (1+9*(1-p_KorrekturfaktorP(p)) )

          + p_TSGehalt("Zuckerrübenmelasse (FM)") * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,"Zuckerrübenmelasse (FM)",tk)
          * (1+9*(1-p_KorrekturfaktorP(p)) )  ;
```

Equation e_MaxZuckMel: Maximum sugar beet molasses fed to ruminants

The model might select a larger amount of sugar beet molasses concentrate fodder fed to ruminants than oil and grain fodders. To address this, we include a constraint in which we assume that amount of sugar beet molasses concentrate fodder can be fed to ruminants in a quantity that is not more than twice the amount of grain fodder and the oil fodder. The constraint becomes gradually stricter over periods.

Similar to constraint *e_MaxKF_Wiederk*, we include this constraint when analyzing the situations without import of concentrate fodder. Depending on the model settings, the constraint can be (de)activated.

```
e_MaxZuckMel(p,tk)..

          p_TSGehalt("Zuckerrübenmelasse (FM)") * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,"Zuckerrübenmelasse (FM)",tk)
          * (1+9*(1-p_KorrekturfaktorP(p)) )

          =L=

          (
          2 * sum(mnp, p_TSGehalt(mnp) * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,mnp,tk) )

          + sum(oel, p_TSGehalt(oel) * p_TierWiederk(tk)
          * v_FutterFuerTiere(p,oel,tk) )
          ) * 0.06 ;
```

Equation e_HartkaeseschotteAnVeredlung: Maximum Hard Cheese Fodder Fed to Ruminants

The amount of hard cheese fodder fed to ruminants is constrained by the assumption that not more than 10% of total hard cheese fodder quantity should be fed.

Similar to constraints *e_MaxKF_Wiederk* and *e_MaxZuckMel*, we include this constraint when analyzing situations in which there is no import of concentrate fodder, and it can be (de)activated depending on the model settings.

```
e_HartkaeseschotteAnVeredlung(p)..

          sum(tk, p_TierWiederk(tk)
          * v_FutterFuerTiere(p,"Hartkäseschotte (FM)",tk))

          =L=

          0.1 * v_Futtermengen(p,"Hartkäseschotte (FM)") ;
```

4.4 Storage of commodities

The model stores data on the raw crop and livestock products, and processed food and feed commodities over certain periods. Storable products are included in the subsets *lp* (general stocks) and *plp* (compulsory stocks). Storage capacities, in terms of weight, are limited in certain cases—for example, in the case of products that need special warehouses, such as cheese or beef meat. Storable crop products must reach minimum stock levels at the end of the period to ensure sufficient supply until the next harvest. The model considers 13 raw crop products, 3 livestock products, 29 food products and 30 feed products for storage.

The compulsory stocks are calculated separately, which—depending on the scenarios—can be defined in the model as available or unavailable for use in the future periods. The compulsory stocks include the products that are stored on behalf of the Swiss national economic supply. In the current model, we consider two raw crop products (wheat and durum wheat grains), three food products (rice, cooking oil, sugar) and one fodder product (soybean meal/oilcake).

Equation e_la: Compulsory storage balance

The model includes the balance storage of compulsory commodities for all of Switzerland at the end of each period. The balance of compulsory storage of commodities consists of stock level at the end of the previous period (for the first period we consider the actual compulsory stocks) minus commodities that are taken out from stock.

```
e_la(p,plp)..
         p_Bestand_PLP(plp) $ (ord(p)=1)
         + v_Pflichtlagerbestaende(p-1,plp)  $ (ord(p)>1)
         - v_AusPflichtlager(p,plp)
         =E=
         v_Pflichtlagerbestaende(p,plp) ;
```

Equation e_lb: General storage balance

We calculate the balance of storage of commodities at the end of each period. The amount of stored commodities includes the storage of commodities at the end of the previous period (for the first period, we consider the actual stocks) and commodities that are deposited into the storage minus commodities that are taken out from the storage.

```
e_lb(p,lp)..
         p_Bestand_LP(lp) $ (ord(p)=1)
         + v_Lagerbestaende(p-1,lp) $ (ord(p)>1)
         + v_InLager(p,lp) - v_AusLager(p,lp)
         =E=
         v_Lagerbestaende(p,lp) ;
```

Equation e_lc: Minimum storage amount

To avoid unnecessary stock input and stock output flows, the quantity of the accumulated storage commodity at the end of the period should be at least the quantity that is deposited into the storage during the period.

```
e_lc(p,lp)..
         v_Lagerbestaende(p,lp)
         =G=
         v_InLager(p,lp) ;
```

Equation e_ld: Storage capacity

Certain commodities have a certain storage capacity, including beef meat and hard/semi-hard cheese. The model restricts the storage amount of these products in accordance with the storage capacity of their warehouses (e.g., cold storage warehouses and cheese storage facilities).

```
e_ld(p,ist) $ (p_Kapazitaet_IST(ist)>0)..
         sum(lp, v_Lagerbestaende(p,lp) $ Lagerkapazitaeten(lp,ist))
         =L=
         p_Kapazitaet_IST(ist) ;
```

Equation e_LagerConsProdukt: Restriction of storage quantity of raw crop products

We assume that the storage of domestically produced raw products is constrained, where the storage quantity of a raw product cannot be larger than the quantity of the domestic production of this raw product. This constraint limits the accumulation of storable products over several periods. It also restricts the storage quantity of imported raw products.

```
e_LagerConsProdukt(p,rp)..

          sum(lp, v_Lagerbestaende(p,lp) $ sameas(rp,lp))

          =L=

          v_InlandPrdmengen(p,rp) ;
```

Equation e_LagerConsNM: Restriction of storage quantity of food products

Similar to the raw products, we limit the storage quantities of processed food products. For this, we assume that the storage quantity of food products cannot be larger than the quantity of domestic production of these food products. This constraint also restricts the storage of imported food products.

```
e_LagerConsNM(p,nm)..

          sum(lp, v_Lagerbestaende(p,lp) $ sameas(nm,lp))

          =L=

          v_AngebotNM(p,nm) ;
```

Equation e_LagerConsFM: Restriction of storage quantity of fodder products

Similar to constraints *e_LagerConsProdukt* and *e_LagerConsNM*, we limit the quantity of stored feed. Therefore, the storage quantity of feed products cannot be larger than the quantity of domestic production of feed products. This constraint also restricts the storage of imported feed products.

```
e_LagerConsFM(p,fm)..

          sum(lp, v_Lagerbestaende(p,lp) $ sameas(fm,lp))

          =L=

          v_AngebotFM(p,fm) ;
```

4.5 Processing and trade

Processing is required to obtain food and feed products from raw agricultural products. For example, grains are processed into baking flour, bread, pastries, and wheat feed; rapeseed into cooking oil; and sugar beet into sugar, molasses, sugar-sweetened beverages, and sugar beet feed. Many processed commodities also produce by-products that are used as livestock feed or food for consumption. Different raw agricultural products can be processed into different food products. For example, raw milk can be processed into a wide variety of dairy products, including milk, low fat milk, skimmed milk, whole milk powder, skimmed milk powder, unsweetened yogurt, sweetened yogurt, hard/semi-hard cheese, soft cheese, fresh cheese, butter, butter milk, cream, ice cream, whole milk for livestock feed, skimmed milk feed, and skimmed milk powder feed. Technological relationships of processing products are represented in the model. For specific products, the current maximum achievable processing capacities are specified (sugar, edible oil, grains, and feed).

Based on foreign trade statistics, the model also includes the trade quantities of food, feed, and seeds that are usually imported and exported in Switzerland. Depending on scenario settings, assumptions on imports and exports can be adjusted.

Equation e_va: Raw products originating from foreign trade

Six raw products are only imported (intermediate feed, durum wheat, malt, maize seed, sugar beet seed, intermediate feed and wheat pasta middling). The following equation ensures that the domestic production of these commodities is zero.

```
e_va(p,sp)..

         sum(rp, v_InlandPrdmengen(p,rp) $ sameas(sp,rp))

         =E=

         0 ;
```

Equation e_vb: Balance of raw products

Equation *e_vb* accounts for the flow of raw products in each period. It includes the inflow of products coming from domestic production, storage (including compulsory storage) and import, as well as outflow of products for export and storage. The domestic raw product supply also takes into account a fixed quantity of products, which is not modelled as domestic production (in the current version of the model, this includes only intermediate green fodder, which implies fodder that is produced during winter season between the harvest of a main crop in autumn and the seed of the next main crop in spring).

```
e_vb(p,rp)..

         ( v_InlandPrdmengen(p,rp)

         + sum(lp, v_AusLager(p,lp) $ sameas(rp,lp))
         - sum(lp, v_InLager(p,lp) $ sameas(rp,lp))

         + sum(plp, v_AusPflichtlager(p,plp) $ sameas(rp,plp))

         + sum(hp, v_importeTot(p,hp) $ sameas(rp,hp))

         - sum(hp, v_Exporte(p,hp) $ (sameas(rp,hp)
         and p_Produktexporte(p,hp)>0) )

         + sum(fp, p_Angebot_FP(p,fp) $ sameas(rp,fp))   )

         =E=

         v_Produktmengen(p,rp) ;
```

Equation e_vc: Minimum processing quantity of raw products

A certain share of domestically produced raw products must be processed within the same period. This implies that the share of products that can be stored must be lower than 100% minus minimum processing share.

```
e_vc(p,rp) $ (p_MinDirektVerarbeitung_RP(rp)>0)..

         sum(lp, v_InLager(p,lp) $ sameas(rp,lp))

         =L=

         (1-p_MinDirektVerarbeitung_RP(rp)) * v_InlandPrdmengen(p,rp) ;
```

Equation e_vd: Maximum processing quantity of raw products

This equation assumes that a certain minimum share of raw products from domestic production quantities must be stored and it should be larger than a certain maximum processing quantity of raw products.

```
e_vd(p,rp) $ (p_MaxDirektVerarbeitung_RP(rp)<1)..

         sum(lp, v_InLager(p,lp) $ sameas(rp,lp))

         =G=

         (1-p_MaxDirektVerarbeitung_RP(rp)) * v_InlandPrdmengen(p,rp) ;
```

Equation e_ve: Balance of processing of raw products

The total quantity of raw products is utilized during the processing stage. In addition, certain raw products can remain unprocessed and, thus, unused (i.e., variable *v_NichtVerarbeitet*). The raw products can be processed into different processed products, either food or feed products. The processing coefficients are taken into account (quantity of raw products per unit of processed products, *p_Koeffizent_VP*); these coefficients consider processing losses, drying, or adding water for products.

```
e_ve(p,ap) $ sum(vp, vb(ap,vp))..

         sum(rp, v_Produktmengen(p,rp) $ sameas(rp,ap)) $ sum(vp, vb(ap,vp))

         =E=

         sum(vp, p_Koeffizient_VP(ap,vp) * v_VerarbPrdMengen(p,ap,vp)
         $ vb(ap,vp))

         + v_NichtVerarbeitet(p,ap) $ sum(vp, vb(ap,vp)) ;
```

Equation e_ve_nv: Unprocessed products

Depending on the scenario, model results may lead to the production of raw products that are not used. This equation ensures that the quantity of unused raw products is zero, which implies that all produced raw products have to be processed. Exceptions are set for three products—fodder straw, offal, and bones—which can be used for non-food processes that are not included in the model.

```
e_ve_nv(p,ap) $ (not sameas(ap,"Futterstroh (RP)")
         and (not sameas(ap,"Innereien (RP)"))
         and (not sameas(ap,"Knochen (RP)")) )..

         v_NichtVerarbeitet(p,ap)

         =E=

         0 ;
```

Equation e_vg: Maximum quantity of processed products

The raw products can be processed into different processed products (see equation *e_ve*). In certain cases, the share of a raw product that is processed to a certain processed product should not exceed a maximum value (parameter *p_MaximalAnteil*), for example, due to technical capacity reasons during processing.

```
e_vg(p,ap,vp) $ (p_MaximalAnteil(ap,vp)<1)..

         p_MaximalAnteil(ap,vp)
         * sum(rp, v_Produktmengen(p,rp) $ sameas(ap,rp)) $ vb(ap,vp)

         =G=

         p_Koeffizient_VP(ap,vp) * v_VerarbPrdMengen(p,ap,vp) $ vb(ap,vp) ;
```

Equation e_vf: Minimum quantity of processed products

Similar to the constraint *e_vf*, the share of a raw product that is processed to a certain processed product should be at least the amount of a minimum value (parameter *p_MinimalAnteil*).

```
e_vf(p,ap,vp) $ (p_MinimalAnteil(ap,vp)>0)..

         ( p_MinimalAnteil(ap,vp)
         * sum(rp, v_Produktmengen(p,rp) $ sameas(ap,rp)) ) $ vb(ap,vp)

         =L=

         (p_Koeffizient_VP(ap,vp) * v_VerarbPrdMengen(p,ap,vp) ) $ vb(ap,vp) ;
```

Equation e_vi: The processing capacity of raw products

The processing capacity of raw products limits the quantity of products that can be processed. Processing capacities are only set for certain raw products, for example, due to required technological installations that cannot be implemented in the short term (e.g., sugar factories for sugar beets, oil mills for oilseeds). There is an additional equation for processing capacities that limits the processing quantities measured in tons of processed products (see *e_vt*).

```
e_vi(p,ist)..

         sum(vb(ap,vp), p_Koeffizient_VP(ap,vp)
         * v_VerarbPrdMengen(p,ap,vp) $ (VerarbKapazitaeten(ap,vp,ist)
         and vb(ap,vp)) )

         =L=

         p_Kapazitaet_IST(ist) ;
```

Equation e_vj: Outputs of by-products

This equation calculates the quantities of by-products for those processed products that lead to more than one final product. We calculate by-product quantities using the coefficient (*p_Koeffizient_KP*) of by-product quantity from the processed (main) products.

```
e_vj(p,ap,vp,kp)..

         ( (v_VerarbPrdMengen(p,ap,vp) $ vb(ap,vp))
         / p_Koeffizient_KP(ap,vp,kp) ) $ vbk(ap,vp,kp)

         =E=

         v_KoppelPrdMengen(p,ap,vp,kp) $ vbk(ap,vp,kp) ;
```

Equation e_vl: Domestic fodder supply

The fodder supply from domestic production consists of those fodder products that are produced as main products and those that are produced as by-products.

```
e_vl(p,fm)..

         sum((ap,vp), v_VerarbPrdMengen(p,ap,vp)
         $ (vb(ap,vp) and sameas(fm,vp)) )

         + sum((ap,vp,kp) $ (vbk(ap,vp,kp) and sameas(fm,kp)),
         v_KoppelPrdMengen(p,ap,vp,kp))

         =E=

         v_AngebotFM(p,fm) ;
```

Equation e_vm: Balance of feed available for livestock

The feed quantities available for livestock are calculated by including domestic production, changes in feed stock, and foreign trade. The feed supply also considers a fixed quantity of products (i.e., *p_Angebot_FP* that currently includes calcium carbonate).

```
e_vm(p,fm)..

         v_AngebotFM(p,fm)

         + sum(lp, v_AusLager(p,lp) $ sameas(fm,lp))
         - sum(lp, v_InLager(p,lp) $ sameas(fm,lp))
         + sum(plp, v_AusPflichtlager(p,plp) $ sameas(fm,plp))

         + sum(hp, v_importeTot(p,hp) $ sameas(fm,hp))
         - sum(hp, v_Exporte(p,hp) $ (sameas(fm,hp)
         and (p_Produktexporte(p,hp)>0)) )

         + 0.6 * sum(fp, p_Angebot_FP(p,fp) $ sameas(fm,fp))

         =E=

         v_Futtermengen(p,fm) ;
```

Equation e_vn: Minimum quantity of available feed

We assume that a minimum quantity of a certain type of feed (particularly preserved roughage) has to be used for the processing in each period. This implies that the share of processed feed that can be stored must be lower than 100% minus this minimum utilization share.

```
e_vn(p,fm) $ (p_MinDirektVerfuegbar_FM(fm)>0)..

         sum(lp, v_InLager(p,lp) $ sameas(fm,lp))

         =L=

         (1-p_MinDirektVerfuegbar_FM(fm)*p_Tage(p)/365)*v_AngebotFM(p,fm) ;
```

Equation e_vo: Maximum quantity of available feed

Maximum utilization of processed feed within the same period assumes that a corresponding minimum share of processed feed must be stored.

```
e_vo(p,fm) $ (p_MaxDirektVerfuegbar_FM(fm)<1)..

         sum(lp, v_InLager(p,lp) $ sameas(fm,lp))

         =G=

         (1-p_MaxDirektVerfuegbar_FM(fm)*p_Tage(p)/365)*v_AngebotFM(p,fm) ;
```

Equation e_Kraftfuttermenge: Domestic concentrate feed supply

In this equation, we calculate the domestic supply amount of concentrate feed supplied for livestock. The amount of concentrate feed is given in its dry matter content. The coefficient 0.001 converts tons into kilotons.

```
e_Kraftfuttermenge(p)..

         0.001 * sum(fm, p_Kraftfutter(fm) * p_TSGehalt(fm)
         * v_AngebotFM(p,fm))

         =E=

         v_Kraftfutterproduktion(p) ;
```

Equation e_AngebotImportFFMenge: Import of concentrate feed

Concentrate feed is also imported from abroad. Here, we calculate the amount of imported concentrate feed in its dry matter content. The coefficient 0.001 converts tons into kilotons. The calculated value is used for the estimation of the Swiss net self-sufficiency rate, which excludes those livestock products that are based on imported feed from the domestic production (see equation *e_SelbstversorgungsgradNetto*).

```
e_AngebotImportFMMenge(p)..

         0.001 * sum(fm, p_Kraftfutter(fm) * p_TSGehalt(fm)
         * sum(hp, v_ImporteTot(p,hp) $ sameas(hp,fm)) )

         =E=

         v_ImportFM(p) ;
```

Equation e_vp: Domestic food supply

Domestic food supply for consumption consists of processed and by-product foods (calculation is similar to the feed supply in equation *e_vl*). In addition, in certain model scenarios that assume a certain reduction of the avoidable food waste, we include a variable (*v_VermVerlustProd*) of additional food supply as a result of the avoided food waste during agricultural production and processing (see also equation *e_FoodWasteProduktion*).

```
e_vp(p,nm)..

         sum((ap,vp), v_VerarbPrdMengen(p,ap,vp)
         $ (vb(ap,vp) and sameas(nm,vp)) )

         + sum((ap,vp,kp), v_KoppelPrdMengen(p,ap,vp,kp)
          $ (vbk(ap,vp,kp) and sameas(nm,kp)) )

         + v_VermVerlustProd(p,nm)

         =E=

         v_AngebotNM(p,nm) ;
```

Equation e_vq: Balance of food products

This equation shows the inflow and outflow of food products from different sources of the food system. The quantity of available food products for consumption comprises of domestic supply, stored products (including storing in storage and compulsory storage, and withdrawing from storage), foreign trade, and processed food products that are supplied in fixed quantity in each year (i.e., *p_Angebot_FP* that includes honey, nuts, game, fish, and fresh vegetables).

```
e_vq(p,nm)..

          v_AngebotNM(p,nm)

          + sum(lp, v_AusLager(p,lp) $ sameas(nm,lp))
          - sum(lp, v_InLager(p,lp) $ sameas(nm,lp))
          + sum(plp, v_AusPflichtlager(p,plp) $ sameas(nm,plp))

          + sum(hp, v_ImporteTot(p,hp) $ sameas(nm,hp))
          - sum(hp, v_Exporte(p,hp) $ (sameas(nm,hp)
            and (p_Produktexporte(p,hp)>0))  )

          + sum(fp, p_Angebot_FP(p,fp) $ sameas(nm,fp))

          =E=

          v_Nahrungsmengen(p,nm) ;
```

Equation e_vr: Minimum Quantity of available food products

A certain share of domestic food products should be utilized during processing. This implies that the share of food products that can be stored must be lower than 100% minus this minimum utilization share (similar equation is used for livestock feed in constraint *e_vn*). In the present version of the model, this constraint is relevant only for canned vegetables.

```
e_vr(p,nm) $ (p_MinDirektVerfuegbar_NM(nm)>0)..

          sum(lp, v_InLager(p,lp) $ sameas(nm,lp))

          =L=

          (1-p_MinDirektVerfuegbar_NM(nm) * p_Tage(p)/365)*v_AngebotNM(p,nm) ;
```

Equation e_vs: Maximum quantity of available food products

The maximum utilization of food products within the same period assumes that a corresponding minimum share of food products must be stored (similar equation is used for livestock feed in constraint *e_vo*).

```
e_vs(p,nm) $ (p_MaxDirektVerfuegbar_NM(nm)<1)..

          sum(lp, v_InLager(p,lp) $ sameas(nm,lp))

          =G=

          (1-p_MaxDirektVerfuegbar_NM(nm) * p_Tage(p)/365)*v_AngebotNM(p,nm) ;
```

Equation e_vt: Processing capacity

The constraint includes a processing capacity for processing operations of raw products (currently 11 crop products) into various processed products (currently eight processed products) over simulated timeframe. In the current version of the model, we assume that processing capacities do not change over time due to the difficulty of increasing them within a short time. The parameter *p_VerarbeitungsKapazitaet* includes the maximum processing capacities measured in tons of processed products per month (the parameter is converted to the duration of the period within the equation). This equation has to be fulfilled in addition to the equation *e_vi* in which the processing capacities are limited in relation to the quantity of raw products instead of processed products.

```
e_vt(ap,vp,pp) $ (p_VerarbeitungsKapazitaet(ap,vp))..

        v_VerarbPrdMengen(pp,ap,vp) $ vb(ap,vp)

        =L=

        p_VerarbeitungsKapazitaet(ap,vp) * p_Tage(pp)/365 * 12 ;
```

Equation e_ExportZuck: Export and import of sugar and sugar-sweetened beverages

The domestic production of sugar and sugar-sweetened beverages can become substantially large in the model optimization process due to their high calories and net revenues. Moreover, the exports of sugar can become too large because it can result in high net revenues and calorie food products. To address these possible high values in sugar production and import, we assume that the export of sugar and 9.5% of sugar-sweetened beverages should not be larger than the import of sugar. This constraint is mainly used in certain scenarios—for example, in scenarios that analyze domestic food self-sufficiency.

```
e_ExportZuck(p)..

        v_ImporteTot(p,"Zucker (NM)")

        =G=

        v_Exporte(p,"Zucker (NM)") + 0.095 * v_Exporte(p,"Süssgetränke (NM)") ;
```

Equation e_ImportZuck: Import of sugar beet fodder

It is assumed that sugar beet fodder import will not increase over time. Accordingly, here, the constraint considers that the import of sugar beet fodder should not be larger than its current import quantity.

```
e_ImportZuck(p)..

        v_importeTot(p,"Zuckerrüben (FM)")

        =L=

        p_Produktimporte(p,"Zuckerrüben (FM)") ;
```

Equation e_ExpSus: Export of sweet products

Sweet products such as sugar, chocolate, cocoa, tropical fruits, other sugar products and other fruit juices have been exported over several years from Switzerland. We assume that the export quantities of these sweet products should not be lower than their current export quantities.

```
e_ExpSus(p,hp) $ (p_ExpSus(hp)>0)..

        p_ExpSus(hp) * v_Exporte(p,hp)

        =G=

        p_ExpSus(hp) * p_Produktexporte(p,hp) ;
```

Equation e_ImportAnteil: Import of vegetables

We assume that a certain share of consumed vegetables and fruits should be imported because not all consumed vegetables and fruits can be produced in Switzerland. The shares of consumed imported vegetables and fruits are derived from the current share of imports of vegetables and fruits relative to their consumption quantity (in certain cases, the share is higher than 1 because a part of the imports is not consumed domestically and instead re-exported). These shares remain constant in the model.

```
e_ImportAnteil(p,hp) $ (p_AnteilObsGem(hp)>0)..

          v_importeTot(p,hp)

          =G=

          p_AnteilObsGem(hp) * sum(nm, v_Nahrungsmengen(p,nm) $ sameas(nm,hp));
```

Equation e_ImportDuer: Import of dried fodder

The import of dried fodder for balanced mixed stock is restricted and expected not to increase over time. In this constraint, we assume that the import of dried fodder for balanced mixed stock not to be larger than its current import quantity.

```
e_ImportDuer(p)..

          v_importeTot(p,"Dürrfutter - Ausgewogener Mischbestand (FM)")

          =L=

          p_Produktimporte(p,"Dürrfutter - Ausgewogener Mischbestand (FM)") ;
```

Equation e_NMRationenGemSum: Total consumption of vegetables

The total consumption quantity of vegetables includes the summed consumption of lettuce; fresh vegetables; and canned fresh, stored, and frozen vegetables. This equation derives a variable (*v_NMRationenGemSum*) for use in constraint *e_GemVer*.

```
e_NMRationenGemSum(p)..

           sum(nm, v_NMRationen(p,nm) * p_ConGem(nm))

           =E=

           v_NMRationenGemSum(p) ;
```

Equation e_GemVer: Minimum consumption of vegetables

We assume that the consumption of vegetables (i.e., lettuce; fresh vegetables; and canned fresh, stored, and frozen vegetables) should be larger than half the share of their consumption relative to the current consumption amount. This equation and equation *e_NMRationenGemSum* are used to ensure a certain variety in the consumption of vegetables. The equations can be (de)activated depending on the scenario settings.

```
e_GemVer(p,nm) $ (p_ConGem(nm)>0)..

          v_NMRationen(p,nm)

          =G=

          ( p_AktuelleRation(nm) / p_SumAktuelleRationGem )
          * v_NMRationenGemSum(p) * 0.5 ;
```

Equation e_FoodWasteProduktion: Reduction of food waste from the processing of products

The food waste during the agricultural production and processing of products can be reduced over the modelled periods due to exogeneous technological developments or changes in production flows (e.g., 75% reduction of the avoidable food waste). This equation is used to analyze the impacts of improvements in reducing the food waste from processing and, as a result, an increase in the supply of food products (see equation *e_vp*).

Accordingly, the equation can be switched on/off depending on whether the reduction in food waste from processing is modelled.

```
e_FoodWasteProduktion(p,nm)..

          v_VermVerlustProd(p,nm)

          =E=

          0.75 * p_KorrekturfaktorP(p) * (p_AvFWProd(nm)/(1 - p_AvFWProd(nm)))
          * v_AngebotNM(p,nm) ;
```

4.6 Food supply self-sufficiency

In this section, we present equations written on production, consumption, import and export of food, as well as domestic food production self-sufficiency. All these equations are calculated in terms of calories and are needed to derive the domestic food supply self-sufficiency based on the methodology used in Switzerland (Agristat, 2024).

Equation e_ProduktionKalorien: Production of domestic food calories

Production of food calories include foods from domestic production and assumed foods that have fixed supply, such as wild animal meat, fish, nuts, honey, and fresh vegetables. Coefficients 0.01 and 0.001 convert to 100 and kiloton calories, respectively.

```
e_ProduktionKalorien(p)..

          0.01 * 0.001 * sum(nm, p_NM_Kaloriengehalt(nm)
          * (v_AngebotNM(p,nm) + sum(fp, p_Angebot_FP(p,fp) $ sameas(fp,nm))) )

          =E=

          v_Kalorienproduktion(p) ;
```

Equation e_VerbrauchKalorien: Consumption of food calories

The consumption of food calories for the entire population is calculated by including total food supply and multiplying it with the calorie content of each food. Coefficients 0.01 and 0.001 convert to 100 and kiloton calories, respectively.

```
e_VerbrauchKalorien(p)..

          0.01 * 0.001 * sum(nm, p_NM_Kaloriengehalt(nm)
          * v_Nahrungsmengen(p,nm))

          =E=

          v_Kalorienverbrauch(p) ;
```

Equation e_ImportKalorien: Import of food calories

The import of food calories is calculated as total food import quantity multiplied by the calorie content of food products. Coefficients 0.01 and 0.001 convert to 100 and kiloton of calories, respectively.

```
e_ImportKalorien(p)..

         0.01 * 0.001 * sum((hp,nm), p_NM_Kaloriengehalt(nm)
         * v_ImporteTot(p,hp) $ sameas(hp,nm))

         =E=

         v_Kalorienimport(p) ;
```

Equation e_ExportKalorien: Export of food calories

Similarly, we calculate the export of food calories by taking the total food export quantity multiplied by the calorie content of food products. Coefficients 0.01 and 0.001 convert to 100 and kiloton of calories, respectively.

```
e_ExportKalorien(p)..

         0.01 * 0.001 * sum((hp,nm), p_NM_Kaloriengehalt(nm)
         * v_Exporte(p,hp) $ sameas(hp,nm))

         =E=

         v_Kalorienexport(p) ;
```

Equation e_SelbstversorgungsgradNetto: Food self-ufficiency

Food self-sufficiency reflects the amount of food, in terms of calories, that is produced domestically. We assume that the domestic food calorie production should be at least 50% of food calorie consumption, which should be achieved over time. Domestic food calorie production is less the import of fodder. Coefficient 0.45 reveals the current food self-sufficiency percentage of total food consumption, while coefficient 0.05 represents the percentage that needs to be increased to reach the minimum food self-sufficiency ratio.

We include this constraint when we analyze the objective of increasing domestic food production. Hence, the constraint can be switched on/off or the food self-sufficiency target can be reduced/increased depending on whether the food self-sufficiency goal is employed and the settings of the scenario analysis.

```
e_SelbstversorgungsgradNetto(p)..

         v_Kalorienproduktion(p) - v_ImportFM(p) * (3052/974)/4.184

         =G=

         (0.45 + 0.05 * p_KorrekturfaktorP(p) ) * v_Kalorienverbrauch(p) ;
```

4.7 Food consumption

The model optimizes agricultural production and the product flows in the food system in a manner that the food consumption and calorie requirements of the Swiss population are satisfied. Calorie requirements are included through the current average calorie consumption per person. We include upper and lower boundary consumption of food groups. The supply of drinking water is not modeled.

To evaluate the nutrient density of foods and diets, we use the Nutrient Rich Food Index 10.3 (NRF10.3; Drewnowski et al., 2021; Fulgani et al., 2019). We include 13 nutrients in food products. Sugar, sodium, and fat are disqualifying nutrients under maximum reference values and contribute negatively to NRF10.3. The remaining 10 nutrients are qualifying nutrients and include protein, fiber, vitamin A, vitamin C, vitamin E, potassium, calcium, magnesium, iodine, and iron. The consumption of these 10 qualifying nutrients under dietary reference intakes contribute positively to an individual's diet. The difference between the sum of qualifying nutrients' consumption and the sum of disqualifying nutrients' consumption yields NRF10.3. For analysis at the diet level, capping is applied at the maximum level of consumption of qualifying nutrients because an increase in the consumption of qualifying nutrients does not lead to an increase in the nutritional proprieties pf a diet. For disqualifying nutrients at the diet level, we assume that these nutrients are only considered when they exceed their highest level of consumption recommendations.

We also modeled the impact of food consumption on the health of the Swiss population. For this, we utilized the Health Nutritional Index (HENI), as described by Stylianou et al. (2021) and based on the epidemiological data from the Global Burden of Disease (2020). In the model, 15 dietary risk factors and alcohol consumption were considered for the HENI analysis. The HENI considers the micro-disability adjusted life years (µDALYs), which is a measure expressed in the number of years lost due to ill-health, disability, or early death as a result of an unhealthy diet.

The modeled food energy supply cannot be directly compared with the energy input because part of the food losses occurring in the value chain are not considered. We include the food waste from agricultural production and from the processing of products. We assume that no food waste occurs during the storage of commodities and during food distribution.

Equation e_ea: Food consumption per capita and day

Daily per capita food consumption comes from processed products. The coefficient 1,000,000 in equation *e_ea* converts tons into grams.

```
e_ea(p,nm)..

         1000000 * v_Nahrungsmengen(p,nm) / (p_Tage(p) * p_Bevoelkerung(p))

         =E=

         v_NMRationen(p,nm) ;
```

Equation e_ey4_Kalorien: Calorie requirement

The actual daily calorie consumption per person that needs to be fulfilled is 2,330, after deducting the food waste. The calorie requirement levels can be increased or reduced depending on scenario settings. Moreover, additional constraints with minimum and maximum consumption of calorie levels can be included.

```
e_ey4_Kalorien(p)..

         v_KalorienAngebotVz(p) / 2330

         =E=

         p_Relax1 ;
```

Equation e_Verzehr: Food consumption amount after deducting food loss

The population does not consume all supplied food products and part of the food is wasted. We assume that food waste originates from household consumption and food disposed into garbage. We do not consider food waste from the storage of food and during food distribution.

```
e_Verzehr(p,nm)..

         (1 - p_FaktorAbfall(nm) - p_FaktorHaushalt(nm)) * v_NMRationen(p,nm)

         =E=

         v_NMVerzehr(p,nm) ;
```

Equation e_VerzehrSce: Amount of food consumption with a reduction in food loss

In contrast to the above equation *e_Verzehr*, here, food waste from consumption decreases over time as a result of improvements that reduce food waste. Similarly, we assume that food waste occurs at the food consumption stage. This equation is utilized when analyzing scenarios with reducing food waste due to technological or behavioral changes. A coefficient of 0.75 indicates a reduction in food waste by 75%.

```
e_VerzehrSce(p,nm)..

         (1 - p_FaktorAbfall(nm) - p_FaktorHaushalt(nm) * (1 - 0.75
         * p_KorrekturfaktorP(p)) ) * v_NMRationen(p,nm)

         =E=

         v_NMVerzehr(p,nm) ;
```

Equation e_EDa: Consumption of food groups

The consumption of food groups includes food products that are consumed less the food waste. The calculation of food waste is included in equation *e_Verzehr*. A coefficient of 1,000 converts the consumption of food groups into grams.

```
e_EDa(p,nmg)..

         sum(nm, p_NMG_Inhalt(nm,nmg)/1000 * v_NMVerzehr(p,nm) )

         =E=

         v_NMGRationen(p,nmg) ;
```

Equation e_EDaSce: Consumption of food groups with food loss reduction

Equation *e_EDaSce* reveals the consumption of food groups with the reduction of food waste due to technological improvements or behavioral changes. A coefficient of 1,000 converts the consumption of food groups into grams.

```
e_EDaSce(p,nmg)..

         sum(nm, (1 - p_FaktorAbfall(nm) - p_FaktorHaushalt(nm)
         * (1 - 0.75 * p_KorrekturfaktorP(p)) ) * p_NMG_Inhalt(nm,nmg)/1000
         * v_NMRationen(p,nm)  )

         =E=

         v_NMGRationen(p,nmg) ;
```

Equation e_EDb: Consumption of nutrients

The consumption of different foods provides different nutrients. Food nutrients include the consumption of nutrients less food waste. A coefficient of 100 converts the nutrient consumption amount into grams.

```
e_EDb(p,nms)..

         sum(nm, p_NMS_Inhalt(nm,nms)/100 * v_NMVerzehr(p,nm) )

         =E=

         v_NMSRationen(p,nms) ;
```

Equation e_EDbSce: Consumption of nutrients with food loss reduction

Here, we consider a gradual decrease in food waste over periods for which the consumption of nutrients is calculated. We use this equation to analyze scenarios with food waste reduction that takes place due to technological changes or environmental policies.

```
e_EDbSce(p,nms)..

          sum(nm, (1 - p_FaktorAbfall(nm) - p_FaktorHaushalt(nm)
          * (1 - 0.75 * p_KorrekturfaktorP(p)) ) * p_NMS_Inhalt(nm,nms)/100
          * v_NMRationen(p,nm)  )

          =E=

          v_NMSRationen(p,nms) ;
```

Equation e_EDc: Consumption of calories

This equation considers the food calorie supply less the food waste that occur during food disposal to a bin and from household consumption (see equation *e_Verzehr*).

```
e_EDc(p)..

          sum(nm, p_NM_Kaloriengehalt(nm)/100 * v_NMVerzehr(p,nm) )

          =E=

          v_KalorienAngebotVz(p) ;
```

Equation e_EDcSce: Consumption of calories with food loss reduction

Here, we calculate the consumption of calories and assume that food waste reduces over the periods of the model as a result of improvements in food waste reduction. Similar to other equations where food consumption waste reduces, this equation is used to analyze scenarios that include a reduction in food waste due to technological changes or environmental policies.

```
e_EDcSce(p)..

          sum(nm, (1 - p_FaktorAbfall(nm) - p_FaktorHaushalt(nm)
          * (1 - 0.75 * p_KorrekturfaktorP(p)) ) * p_NM_Kaloriengehalt(nm)/100
          * v_NMRationen(p,nm) )

          =E=

          v_KalorienAngebotVz(p) ;
```

Equation e_EY7nmg1: Minimum consumption of food groups

We assume that each person has a consumption demand of food products. We define the consumption demand of food products by considering the consumption requirement for different food groups. Food groups include different food products, where these food products can substitute the consumption of other food products in the food group. The consumption of food groups per person should not be lower than the defined consumption amount of food groups that remains constant over periods.

```
e_EY7nmg1(p,nmg) $ (p_NMGVzMin(nmg)>0)..

         v_NMGRationen(p,nmg)

         =G=

         p_NMGVzMin(nmg) ;
```

Equation e_EY7nmg2: Maximum consumption of food groups

In this constraint, we define the maximum consumption of food groups per person, where the consumption of food groups should be equal to or lower than the consumption of a certain amount of food groups that remains constant over periods.

```
e_EY7nmg2(p,nmg) $ (p_NMGVzMax(nmg)>0)..

         v_NMGRationen(p,nmg)

         =L=

         p_NMGVzMax(nmg) ;
```

Equation e_EY7nmg1Sce: Minimum consumption of food groups according to the food pyramid

This and subsequent equations (i.e., *e_EY7nmg2Sce*), assume that food consumption has to improve over time in accordance with the Swiss food pyramid. In this constraint, we assume that the consumption of food groups per person should not be lower than the Swiss food pyramid recommendations regarding the minimum consumption amount of food groups. The minimum consumption requirement according to the Swiss food pyramid is assumed to be adopted over periods in the model and substitutes the current consumption of food groups in the last period of the model. Equations regarding the Swiss food pyramid are employed in scenarios that analyze the improvement of food consumption based on recommendations.

```
e_EY7nmg1Sce(p,nmg) $ (p_NMGVzMin(nmg)>0)..

         v_NMGRationen(p,nmg)

         =G=

         ( (1 - p_KorrekturfaktorP(p)) * p_NMGVzMin(nmg)
         + p_KorrekturfaktorP(p) * p_NMGVzGsMin(nmg) ) ;
```

Equation e_EY7nmg2Sce: Maximum consumption of food groups according to the food pyramid

Consumption of food groups per person should be equal or lower than the Swiss food pyramid recommendations on the maximum consumption amount of food groups. The maximum consumption demand according to Swiss food pyramid is implemented over periods and substitute current consumption of food groups in the last period of the model.

```
e_EY7nmg2Sce(p,nmg) $ (p_NMGVzMax(nmg)>0)..

         v_NMGRationen(p,nmg)

         =L=

         ( (1 - p_KorrekturfaktorP(p)) * p_NMGVzMax(nmg)
         + p_KorrekturfaktorP(p) * p_NMGVzGsMax(nmg) )  ;
```

Equation e_ee: Maximum consumption of food products

To avoid a large increase in the consumption of certain products in contrast to the current situation, the model assumes that the consumption of food products per person cannot be larger than five times their consumption amount. This constraint is used in scenarios that simultaneously model several policies, such as an increase in domestic food self-sufficiency, decrease in environmental impacts, and food consumption recommendation. This restriction remains constant over the years.

```
e_ee(p,nm)..

         v_NMRationen(p,nm)

         =L=

*        500% Erhöhung die Aktuelle Nahrungsrationen
         5 * p_AktuelleRation(nm) ;
```

Equation e_MaxSomeNN: Maximum consumption of certain food products

Similar to the previous equation *e_ee*, we also include a constraint that restricts a large increase in the consumption of certain products in contrast to the current situation. The model assumes that the consumption of cooking oil, sugar, sugar-sweetened beverages, molasses, pastry, honey, and sugar-sweetened yoghurt cannot be larger than certain assumed values. The consumption of these products is restricted because they have a high sugar and calorie content. The constraint is used in scenarios where the model maximizes calorie consumption levels. The constraint remains constant over the years.

```
e_MaxSomeNM(p,nm) $ (p_MaxSomeNM(nm)>0)..

         v_NMRationen(p,nm)

         =L=

         p_MaxSomeNM(nm) ;
```

Equation e_RationNebenprodukte1: Maximum consumption of molasses

We assume that consumption of molasses cannot exceed a certain level because this product has a high calorie content; thus, it might be consumed in excessively large quantities to meet the calorie requirement constraint of the model (i.e., equation *e_ey4_Kalorien*).

```
e_RationNebenprodukte1(p,"Melasse (NM)")..

         v_NMRationen(p,"Melasse (NM)")

         =L=

         10 ;
```

Equation e_RationNebenprodukte2: Maximum consumption of malt

Similar to the previous constraint, we assume that the consumption of malt cannot exceed a certain level due to its high calorie content and possibly the excessive consumption amount selected by the model.

```
e_RationNebenprodukte2(p,"Malz (NM)")..

          v_NMRationen(p,"Malz (NM)")

          =L=

          30 ;
```

Equation e_RationPlusTierfett: Maximum consumption of animal fat

We assume that consumption of animal fat cannot be more than twice the amount of its actual (current) consumption.

```
e_RationPlusTierfett(p,nm)..

          p_RationPlusTierfett(nm) * v_NMRationen(p,nm)

          =L=

          2 * p_RationPlusTierfett(nm) * p_AktuelleRation(nm) ;
```

Equation e_Bier: Maximum consumption of beer

We assume that consumption of beer cannot be more than 1.5 times the amount of its actual (current) consumption. This restriction enables the avoidance of a high increase in beer consumption, as beer has high calorie content; hence the model may yield substantial beer consumption to meet daily calorie requirements.

```
e_Bier(p)..

          v_NMRationen(p,"Bier (NM)")

          =L=

          1.5 * p_AktuelleRation("Bier (NM)") ;
```

Equation e_Spirituosen: Maximum consumption of alcoholic beverages

We assume that the consumption of alcoholic beverages cannot be greater than 1.5 times the amount of its actual consumption. This restriction enables the avoidance of a high increase in the consumption of alcoholic beverages, which have high calorie content.

```
e_Spirituosen(p)..

          v_NMRationen(p,"Spirituosen (NM)")

          =L=

          1.5 * p_AktuelleRation("Spirituosen (NM)") ;
```

Equation e_ApfelW: Maximum consumption of apple wine

We assume that consumption of apple wine is restricted to a maximum of 1.5 times of its actual consumption. This restriction avoids a high increase in apple wine consumption.

```
e_ApfelW(p) ..

          v_NMRationen(p,"Apfelwein (NM)")

          =L=

          1.5 * p_AktuelleRation("Apfelwein (NM)") ;
```

Equation e_Wein: Maximum consumption of wine

Similar to equations *e_Bier*, *e_Spirituosen* and *e_ApfelW*, we assume that consumption of wine cannot be more than 1.5 times the amount of its actual (current) consumption. This restriction enables the avoidance of a high increase in wine consumption to meet calorie requirements.

```
e_Wein(p)..
         v_NMRationen(p,"Wein (NM)")
         =L=
         1.5 * p_AktuelleRation("Wein (NM)") ;
```

Equation e_Backwaren: Maximum consumption of pastries

The consumption of pastries cannot be larger than their current consumption amount. This restriction is needed because pastries have high calories and, as a result, the model can select them in large quantities to meet calorie requirements.

```
e_Backwaren(p)..
         v_NMRationen(p,"Backwaren (NM)")
         =L=
         p_AktuelleRation("Backwaren (NM)") ;
```

Equation e_NMRatMaisgriess: Maximum consumption of cornmeal

We assume the maximum consumption of calories coming from cornmeal cannot be larger than 10% of calories from all the food consumed.

```
e_NMRatMaisgriess(p)..
      p_NM_Kaloriengehalt("Maisgriess (NM)") * v_NMRationen(p,"Maisgriess (NM)")
      =L=
      0.1 * sum(nm, p_NM_Kaloriengehalt(nm) * v_NMRationen(p,nm)) ;
```

Equation e_DN: Consumption of disqualifying nutrients

Food products consist of different nutrients and their consumption can have different nutritional impacts. In this equation, we show the calculation of food consumption of disqualifying nutrients, including salt, sugar, and fat from each consumed food product. These disqualifying nutrients have a negative nutritional impact of consumption.

```
e_DN(p,nm,nms)..
         p_MRV_Inhalt(nm,nms) * v_NMVerzehr(p,nm)
         =E=
         v_DN(p,nm,nms) ;
```

Equation e_LIM: Limiting nutrient index by food product

Equation *e_LIM* shows the limiting nutrient index by food product that considers the same values as the consumption of disqualifying nutrients from each consumed food product.

```
e_LIM(p,nm,nms)..

          v_DN(p,nm,nms)

          =E=

          v_LIM(p,nm,nms) ;
```

Equation e_LIMsum: Sum of limiting nutrient index

This equation shows the per capita daily consumption of disqualifying nutrients from the consumption of all food products.

```
e_LIMsum(p,nms)..

          v_LIMsum(p,nms)

          =E=

          sum(nm, v_LIM(p,nm,nms)) ;
```

Equation e_QN: Consumption of qualifying nutrients

Food consumption results in the consumption of qualifying nutrients of 10 nutrients from each consumed food product. These qualifying nutrients have a positive nutritional impact of consumption.

```
e_QN(p,nm,nms)..

          p_DRI_Inhalt(nm,nms) * v_NMVerzehr(p,nm)

          =E=

          v_QN(p,nm,nms) ;
```

Equation e_NR: Nutrient rich index by food product

Equation *e_NR* shows the nutrient rich index by each food product that considers the same values as a consumption of qualifying nutrients from each consumed food product.

```
e_NR(p,nm,nms)..

          v_QN(p,nm,nms)

          =E=

          v_NR(p,nm,nms) ;
```

Equation e_NRsum: Sum of nutrient rich index

This equation reveals per capita daily consumption of qualifying nutrients (i.e., nutrient rich index). It is calculated as the sum of food consumption for each qualifying nutrient.

```
e_NRsum(p,nms)..

          v_NRsum(p,nms)

          =E=

          sum(nm, v_NR(p,nm,nms)) ;
```

Equation e_NRF: Nutrient Rich Food Index

The Nutrient Rich Food Index (NRF10.3) calculates the difference between the nutrient rich index and the limiting nutrient index for each consumed food product. It reveals the positive or negative nutritional value of consumed food products. The variable *v_NRF* is a free variable and reveals the positive and negative values of food consumption.

```
e_NRF(p,nm,nms)..
          v_NRF(p,nm,nms)
          =E=
          v_NR(p,nm,nms) - v_LIM(p,nm,nms) ;
```

Equation e_HENI: Health Nutritional Index by food product

This equation calculates the daily impact of 1 gm of each consumed food product on a person's health. The results of calculation can be positive (health benefits) and negative (detrimental health effects) values of *v_HENI* variable of each consumed food product.

```
e_HENI(p,nm)..
          p_HENI(nm) * v_NMVerzehr(p,nm)
          =E=
          v_HENI(p,nm) ;
```

Equation e_HENIsum: Health Nutritional Index

Equation *e_HENIsum* presents the overall per capita daily impact on health from the summed consumption of foods. The *v_HENIsum* variable is a free variable. The positive value of *v_HENIsum* variable (which is a HENI index of food consumption) reveals the overall health benefits of food consumption, while the negative value of *v_HENIsum* variable reveals the detrimental impacts of food consumption on health.

```
e_HENIsum(p)..
          sum(nm, v_HENI(p,nm))
          =E=
          v_HENIsum(p) ;
```

4.8 Environmental impacts

The environmental module includes environmental impacts from crop cultivation area, livestock number and manure, processing of raw products, import and export of commodities, and cooking of food.

Equation e_DiffNHof: Nitrogen surplus or shortage

Crops receive predetermined amounts of farm manure and mineral fertilizers. The proportions used to determine the environmental impact of fertilization are set in such a manner that, in the current situation (i.e., initial year of the model), the total requirement for farm manure corresponds to the amount produced by livestock. If the livestock numbers and crop areas change during an optimization, there may be a surplus or shortfall of farm manure in total in relation to the specified farm manure requirement of the crops. These surplus or shortfall amounts of farm manure nutrients are determined using balance equations. Positive differences correspond to a shortfall in the requirement of farm manure nutrients, while negative differences correspond to an increase in the requirement. The balancing of farm manure with mineral fertilizer also leads to a need for correction of environmental effects. The difference variables are linked to an environmental module that includes changes in environmental impacts when 1 kg of farm manure nitrogen (N-available) is replaced by 1 kg of mineral fertilizer nitrogen.

```
e_DiffNHof(p)..

         sum(tk, p_Nabzug(tk) * v_Durchschnittsbestaende(p,tk)
         $ sum(fw, tk_fw(tk,fw)) )

         - sum((ek,inte,eg), p_NKgHa(ek,inte,eg) * p_AnteilHofN(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         =E=

         v_DifferenzNHofPos(p) - v_DifferenzNHofNeg(p) ;
```

Equation e_DiffPHof: Phosphorous surplus or shortage

The net phosphorous supply for agricultural production is calculated in the same manner as it is done for the nitrogen supply in the equation *e_DiffNHof*.

```
e_DiffPHof(p)..

         sum(tk, p_Pabzug(tk) * v_Durchschnittsbestaende(p,tk)
         $ sum(fw, tk_fw(tk,fw)) )

         - sum((ek,inte,eg), p_P2O5KgHa(ek,inte,eg) * p_AnteilHofP(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         =E=

         v_DifferenzPHofPos(p) ;
```

Equation e_TotalBedarfN: Nitrogen requirement

Nitrogen is required for the growth of crops. Hence, total nitrogen use consists of fixed nitrogen application for each crop according to the per hectare nitrogen use requirements for crops.

```
e_TotalBedarfN{p}..

         sum((ek,inte,eg), p_NKgHa(ek,inte,eg) * v_Flaechen(p,ek,inte,eg))

         =E=

         v_BedarfN(p) ;
```

Equation e_TotalBedarfP: Phosphorous requirement

Similar to the previous equation *e_TotalBedarfN*, phosphorous is only used for crop cultivation. Hence, the total phosphorous use consists of the amount of phosphorous application requirements for crops.

```
e_TotalBedarfP(p)..

         sum((ek,inte,eg), p_P2O5KgHa(ek,inte,eg) * v_Flaechen(p,ek,inte,eg))

         =E=

         v_BedarfP(p) ;
```

Equation e_TotalBedarfNHof: Nitrogen requirement from manure

Nitrogen from manure is used for crop cultivation. The amount of nitrogen from manure is calculated as a proportion of the total nitrogen requirement for crops.

```
e_TotalBedarfNHof(p)..

         sum((ek,inte,eg), p_NKgHa(ek,inte,eg) * p_AnteilHofN(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         =E=

         v_BedarfNHof(p) ;
```

Equation e_TotalBedarfPHof: Phosphorous requirement from manure

A certain share of phosphorous livestock manure is also used for crop cultivation. The use of manure phosphorous is governed by the proportion of the total phosphorous requirements for crops.

```
e_TotalBedarfPHof(p)..

         sum((ek,inte,eg), p_P205KgHa(ek,inte,eg) * p_AnteilHofP(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         =E=

         v_BedarfPHof(p) ;
```

Equation e_TotalAnfallNHof: Supply of nitrogen from livestock manure

The nitrogen in manure comes from livestock, which is calculated by including the average number of livestock and manure nitrogen outputs from each livestock type.

```
e_TotalAnfallNHof(p)..

         sum(tk, p_Nabzug(tk)
         * v_Durchschnittsbestaende(p,tk) $ sum(fw, tk_fw(tk,fw)))

         =E=

         v_AnfallNHof(p) ;
```

Equation e_TotalAnfallPHof: Supply of phosphorous from livestock manure

Phosphorous in manure also comes from livestock. The amount of phosphorous in manure is calculated by including the average livestock number and manure phosphorous outputs from each livestock type.

```
e_TotalAnfallPHof(p)..

         sum(tk, p_Pabzug(tk)
         * v_Durchschnittsbestaende(p,tk) $ sum(fw, tk_fw(tk,fw)))

         =E=

         v_AnfallPHof(p) ;
```

Equation e_c1_ek: Environmental impacts of crop cultivation

Each stage of the food system impacts the environment. For example, crop cultivation impacts the environment, including GHG emissions, nitrogen leaching, and other environmental impacts. The environmental impacts of crop cultivation are summed-up over all crops by regions and production methods, and presented as a variable *v_Umwelt_EK*. This variable is free and includes both positive and negative environmental impacts of crop cultivation. The coefficient of 0.001 converts the environmental impact values into tons.

```
e_c1_ek(p,uwa)..

         0.001 * sum((ek,inte,eg), p_UW_EK(ek,inte,eg,uwa)
         * v_Flaechen(p,ek,inte,eg) )

         =E=

         v_Umwelt_EK(p,uwa) ;
```

Equation e_c2_trp: Environmental impacts of livestock number

Rearing livestock leads to environmental impacts. The environmental impacts of livestock are calculated according to the number of livestock and the sum of all livestock. To compensate for any negative values, the variable considers both positive and negative impacts.

```
e_c2_tk(p,uwa)..

         0.001 * sum(tk, p_UW_TK(tk,uwa) * p_FaktorGVE(tk)
         * v_Durchschnittsbestaende(p,tk) )

         =E=

         v_Umwelt_TK(p,uwa) ;
```

Equation e_c3_vrb: Environmental impacts of processed products

Turning raw products into processed products affects the environment. The environmental impacts of processing products are summed up over all processed main products. Similar to the above environmental equations, *v_Umwelt_VRB* variable is free and considers both positive and negative values.

```
e_c3_vrb(p,uwa)..

         sum((ap,vp), p_UW_VRB(ap,vp,uwa) * v_VerarbPrdMengen(p,ap,vp)
         * p_Koeffizient_VP(ap,vp)  $ vb(ap,vp))

         =E=

         v_Umwelt_VRB(p,uwa) ;
```

Equation e_c4_imp: Environmental impacts of imports

Imported products into Switzerland are associated with environmental impacts that occurred abroad during their production and processing stages. However, the environmental impacts of imported products need to be accounted for in the model, because Swiss consumers and other actors in the Swiss food system use these products. The environmental impacts of imported products are summed-up over all imported products and represented in *v_Umwelt_IMP* variable, which is a free variable.

```
e_c4_imp(p,uwa)..

         sum(hp, p_UW_IMP(hp,uwa) * v_ImporteTot(p,hp))

         =E=

         v_Umwelt_IMP(p,uwa) ;
```

Equation e_c5_exp: Environmental impacts of exports

Similar to imported products, exported products also result in environmental impacts. The environmental impacts of exported products include these impacts for all exported products. Similar to the above constraints, we use a free variable that includes positive and negative environmental impacts.

```
e_c5_exp(p,uwa)..

         sum(hp, p_UW_EXP(hp,uwa) * v_Exporte(p,hp)
         $ (p_Produktexporte(p,hp)>0))

         =E=

         v_Umwelt_EXP(p,uwa) ;
```

Equation e_c6_exp: Environmental impacts of manure

Manure production from livestock affects the environment. The environmental impacts of manure production are corrected to align with farm manure accumulation. We compensate for negative values using a second variable that includes only negative values (*v_Umwelt_HFD_N*).

```
e_c6_hfd(p,uwa)..

         (p_UW_HFD(uwa) * v_DifferenzNHofNeg(p)

         - p_UW_HFD_R(uwa) * v_DifferenzNHofPos(p)) * 0.001

         =E=

         v_Umwelt_HFD(p,uwa) - v_Umwelt_HFD_N(p,uwa) ;
```

Equation e_c7_exp: Environmental impacts of cooking food

The model also accounts for the environmental impacts of cooking food products. These impacts are summed up over all cooked food products. Similar to the above constraints, we consider both positive and negative impacts in the free variable.

```
e_c7_zub(p,uwa)..

         sum(nm, p_UW_ZUB(nm,uwa) * v_Nahrungsmengen(p,nm))

         =E=

         v_Umwelt_ZUB(p,uwa) ;
```

Equation e_c8_prp: Environmental impacts of raw crop products

In the previous version of the model (i.e., Green DSS-ESSA model), the environmental impacts of crops were calculated for each raw crop product instead of crop area (e.g., von Ow et al., 2020). The environmental impacts of each raw crop product are calculated based on the output quantity of these products. The current version of the model does not consider this equation. This equation as well as *e_c9_trp, e_c10_rem*, and *e_c11_vrb* equations are listed here to subsequently explain the objective function used to validate the transfer of the model from LPL software into GAMS (see section 7).

```
e_c8_prp(p,uwa)..

          sum((ek_eg_pp), v_Erntemengen(p,ek_eg_pp) * p_UW_PRP(ek_eg_pp,uwa))

          =E=

          v_Umwelt_PRP(p,uwa) ;
```

Equation e_c9_trp: Environmental impacts of raw livestock products

Similar to equation *e_c8_prp*, in the Green DSS-ESSA model, the environmental impacts of each raw livestock product include environmental impacts of the production of milk, eggs, and meat from slaughtered livestock. This equation was used in previous studies (e.g., von Ow et al., 2020) and is used only during the transfer of the model from LPL software into GAMS. Hence, this equation is not used in the current version of the SWISSfoodSys model.

```
e_c9_trp(p,uwa)..

          sum((tk,trp), p_UW_TRP(tk,trp,uwa)
          * v_Tierproduktemengen(p,tk,trp) $ tk_tp(tk,trp))

          + sum((tk,trp), p_UW_TRP(tk,trp,uwa)
          * v_Schlachtproduktemengen(p,tk,trp) $ tk_sp(tk,trp))

          =E=

          v_Umwelt_TRP(p,uwa) ;
```

Equation e_c10_rem: Environmental impacts of livestock replacement by generation

The replacement of livestock by the next generation also has environmental impacts. This equation is not used in the current version of the model because the environmental impacts of livestock are already included in equation *e_c2_trp*. This equation is only used during the model transfer from LPL software into GAMS and the model transfer validation process.

```
e_c10_rem(p,uwa)..

          sum((fg,vg), 0.001 * p_UW_REM(vg,fg,uwa)
          * v_Remonten(p,vg,fg) $ tt(vg,fg))

          =E=

          v_Umwelt_REM(p,uwa) ;
```

Equation e_c11_vrb: Environmental impacts of by-products

In the Green DSS-ESSA model, the processing of by-products also has environmental impacts. However, the current version of the model does not use this equation because the environmental impacts of all processed products are calculated in equation *e_c3_vrb*. This equation is only used during the model transfer from LPL software into GAMS and the model transfer validation process.

```
e_c11_kpd(p,uwa)..

          sum((ap,vp,kp), p_UW_KPD(ap,vp,kp,uwa) * v_KoppelPrdMengen(p,ap,vp,kp)
          $ vbk(ap,vp,kp))

          =E=

          v_Umwelt_KPD(p,uwa) ;
```

Equation e_PSM_OFG: Pesticide risk indicator for surface water

The application of pesticides on crops results in environmental impacts, such as impacts on the surface water, semi-natural habitats, and groundwater. We calculate these impacts using the pesticide risk indicator developed by Korkaric et al. (2023). The pesticide risk indicator for surface water is calculated for each crop based on the method of production. The pesticide risk indicator for semi-natural habitats is calculated by equation *e_PSM_NL*. The pesticide risk indicator for groundwater is calculated by equation *e_PSM_GW*. It must be noted that pesticide risk indicators are not considered for livestock or in other stages of the food system.

```
e_PSM_OFG(p,ek,inte)..

         p_PSM_OFG(ek,inte) * sum(eg, v_Flaechen(p,ek,inte,eg) )

         =E=

         v_PSM_OFG(p,ek,inte) ;
```

Equation e_PSM_NL: Pesticide risk indicator for semi-natural habitats

The pesticide risk indicator for semi-natural habitats is calculated for each crop based on the method of production.

```
e_PSM_NL(p,ek,inte)..

         p_PSM_NL(ek,inte) * sum(eg, v_Flaechen(p,ek,inte,eg) )

         =E=

         v_PSM_NL(p,ek,inte) ;
```

Equation e_PSM_GW: Pesticide risk indicator for groundwater

Similar to the previous equations, the pesticide risk indicator for groundwater is calculated for each crop based on the method of production.

```
e_PSM_GW(p,ek,inte)..

         p_PSM_GW(ek,inte) * sum(eg, v_Flaechen(p,ek,inte,eg) )

         =E=

         v_PSM_GW(p,ek,inte) ;
```

Equation e_IPCCTotal: Balance of GHG emissions in the food system

The different stages of the food production system lead to the production of GHG emissions. The balance of GHG emissions in the food system considers emissions from the entire food system. The GHG emissions are calculated for each stage of the food system, such as that for crop production area, livestock number, processing of raw agricultural products, trade of commodities, cooking of food, and livestock manure supply. We calculate GHG emissions for the inflow and outflow of products into the Swiss food system. When products are exported outside of Switzerland, we assume their GHG emissions during export are relevant to foreign countries and, hence, we subtract export emissions in the equation. The GHG emissions of imported commodities reflect the emissions from producing the imported products that are consumed in Switzerland and, thus, are considered to accumulate emissions in Switzerland. GHG emissions are in the form of CO2 equivalents and represented by the *IPCC 2021 - GWP100 (fossil & LULUC)* set element.

```
e_IPCCTotal(p)..

          v_THGKonsum(p)

          =E=

          0.001 * ( v_Umwelt_EK(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          + v_Umwelt_TK(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          + v_Umwelt_VRB(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          + v_Umwelt_IMP(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          - v_Umwelt_EXP(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          + v_Umwelt_ZUB(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          + v_Umwelt_HFD(p,"IPCC 2021 - GWP100 (fossil & LULUC)")
          - v_Umwelt_HFD_N(p,"IPCC 2021 - GWP100 (fossil & LULUC)") ) ;
```

Equation e_IPCCReduktion: Reduction of GHG emissions in the food system

In certain scenarios, the objective can be to analyze the impacts of reducing GHG emissions on the food system. For such scenarios, we simulate a constraint according to which GHG emissions from the entire food system need to be reduced over the modelled period. We assume that GHG emission reduction has to be achieved per capita, while considering the actual per capita emissions (i.e., here 14249 KtCO2e) and the reduction target level that needs to be achieved (i.e., here 14249 * 0.66 in KtCO2e) over the modeled periods. The GHG reduction target level can be adjusted. Moreover, depending on study objectives, this constraint can be (de)activated.

```
e_IPCCReduktion(p)..

          v_THGKonsum(p) / p_BevoelkerungIndex(p)

          =L=

          14249 - (14249 * 0.66)  * p_KorrekturfaktorP(p) ;
```

Equation e_Treibhausgasemissionen: Balance of agricultural GHG emissions

We also calculate agricultural GHG emissions, which include emissions from crop production area, livestock number, and manure supply. GHG emissions are in the form of CO2 equivalents and represented by the *EmDm - IPCC 2021 - GWP100* set element.

```
e_Treibhausgasemissionen(p)..

          v_THG(p)

          =E=

          0.001 * ( v_Umwelt_EK(p,"EmDm - IPCC 2021 - GWP100")
          + v_Umwelt_TK(p,"EmDm - IPCC 2021 - GWP100")
          + v_Umwelt_HFD(p,"EmDm - IPCC 2021 - GWP100")
          - v_Umwelt_HFD_N(p,"EmDm - IPCC 2021 - GWP100") ) ;
```

Equation e_ReduktionTHG: Reduction of agricultural GHG emissions

In certain modelled scenarios, the objective can be to analyze the impacts of reducing agricultural GHG emissions on the food system. Here, the equation imposes a constraint to reduce agricultural GHG emissions over the modelled period. GHG emission reduction must be achieved for the agricultural production level only, considering current agricultural emissions (i.e., here 6512 KtCO2e) and the reduction target level of 40% that needs to be achieved (i.e., here 6512 – 7678 × 0.6 in KtCO2e) over the modeled periods. Because the target level was set in 1990 and since then agricultural GHG emissions have reduced by 10%, we assume that the GHG reduction target level is 30% of agricultural emissions. The target reduction levels can be adjusted depending on scenario settings; moreover, the constraint can be switched on/off depending on study objectives.

```
e_ReduktionTHG(p)..

         v_THG(p)

         =L=

         6512 - (6512 - 7678 * 0.6) * p_KorrekturfaktorP(p) ;
```

Equation e_Ammoniakemissionen: Ammonia emissions

We also calculate emissions from nitrogen application in agricultural production to analyze the environmental impacts of fertilizer application. These emissions consist of three gases— ammonia, nitrate, and nitrous oxide. In this equation, we calculate ammonia emissions consisting of crop production area, livestock number, and manure supply. The factor of 0.82 is the conversion factor of ammonia to nitrogen, while 0.001 is a conversion factor into kilo tons. Nitrate and nitrous oxide emissions are calculated in equations *e_Nitratemissionen* and *e_Lachgasemissionen*, respectively.

```
e_Ammoniakemissionen(p)..

         0.001 * 0.82 * (v_Umwelt_EK(p,"EmDM - Ammonia")
         + v_Umwelt_TK(p,"EmDM - Ammonia")
         + v_Umwelt_HFD(p,"EmDM - Ammonia")
         - v_Umwelt_HFD_N(p,"EmDM - Ammonia")
         )

         =E=

         v_NH3(p) ;
```

Equation e_Nitratemissionen: Nitrate emissions

Nitrate emissions originate from crop production area, livestock number, and manure supply. The factor of 0.23 is the conversion factor of nitrate to nitrogen.

```
e_Nitratemissionen(p)..

         0.001 * 0.23 * (v_Umwelt_EK(p,"EmDM - Nitrate")
         + v_Umwelt_TK(p,"EmDM - Nitrate")
         + v_Umwelt_HFD(p,"EmDM - Nitrate")
         - v_Umwelt_HFD_N(p,"EmDM - Nitrate")
         )

         =E=

         v_NO3(p) ;
```

Equation e_Lachgasemissionen: Nitrous oxide emissions

Similar to ammonia and nitrate emissions, nitrous oxide emissions originate from crop production area, livestock number, and manure supply. The factor of 0.422 is the conversion factor of nitrous oxide to nitrogen.

```
e_Lachgasemissionen(p)..

         0.001 * 0.64 * (v_Umwelt_EK(p,"EmDM - Nitrous oxide")
         + v_Umwelt_TK(p,"EmDM - Nitrous oxide")
         + v_Umwelt_HFD(p,"EmDM - Nitrous oxide")
         - v_Umwelt_HFD_N(p,"EmDM - Nitrous oxide")
         )

         =E=

         v_N2O(p) ;
```

Equation e_Inertstickstoffemissionen: Inert nitrogen emissions

We also calculate inert nitrogen emissions by combining the emissions of three gases—ammonia, nitrate, and nitrous oxide—and multiplying this by a factor of 0.33.

```
e_Inertstickstoffemissionen(p)..
         0.33 * (v_NH3(p) + v_NO3(p) + v_N2O(p))
         =E=
         v_InertN(p) ;
```

Equation e_OSPARBilanzNV2: Reduction of nitrogen use

In certain scenarios, we model the impacts of reducing nitrogen emissions. In such scenarios, we impose a constraint to reduce nitrogen emissions from agricultural production. Here, we include a constraint on current agricultural nitrogen emissions (i.e., here 97.057 KtN) and the reduction target level of 20% that needs to be achieved (i.e., here 97.057 * 0.2) over the modelled periods. The target reduction levels can be adjusted or deactivated depending on scenario settings.

```
e_OSPARBilanzNV2(p)..
         v_NH3(p) + v_NO3(p) + v_N2O(p) + v_InertN(p)
         =E=
         97.057 - 97.057 * 0.2 * p_KorrekturfaktorP(p) ;
```

Equation e_OsparFutterN: Nitrogen leaching in imported fodder

We calculate nitrogen leaching from different sources to understand nitrogen flows in detail. In this equation, nitrogen leaching is calculated for the imported fodder for livestock. Different sources of nitrogen leaching from agricultural inputs are also calculated by equations *e_OsparDuengerN*, *e_OsparFixierungN*, and *e_OsparDepositionN*. In addition, different sources of nitrogen leaching of food products are calculated by equations *e_OsparEntzugPflanzeN* and *e_OsparEntzugTierN*.

```
e_OsparFutterN(p)..
         0.001 * sum(hp, p_OspNproTo(hp) * v_ImporteTot(p,hp))
         =E=
         v_OsparFuN(p) ;
```

Equation e_OsparFutterP: Phosphorous leaching in imported fodder

Similar to the above equation, we calculate phosphorous leaching for imported fodder for livestock. Different sources of phosphorous leaching from agricultural inputs are also calculated in equations *e_OsparDuengerP* and *e_OsparDepositionP*. The sources of phosphorous leaching related to food products are calculated by equations *e_OsparEntzugPflanzeP* and *e_OsparEntzugTierP*.

```
e_OsparFutterP(p)..

        0.001 * sum(hp, p_OspPproTo(hp) * v_ImporteTot(p,hp))

        =E=

        v_OsparFuP(p) ;
```

Equation e_OsparDuengerN: Nitrogen leaching in mineral fertilizers

Nitrogen leaching from mineral fertilizers is calculated as nitrogen leaching from crop cultivation and current livestock number less the modeled livestock number.

```
e_OsparDuengerN(p)..

        sum((eg,inte,ek), 0.001 * p_OspDueN(ek,inte,eg)
        * v_Flaechen(p,ek,inte,eg))

        + 0.001 * sum(tk, p_Nabzug(tk) * p_AktBestand(tk))

        - 0.001 * sum(tk, p_Nabzug(tk) * v_Tierbestaende(p,tk))

        =E=

        v_OsparDueN(p) ;
```

Equation e_OsparDuengerP: Phosphorous leaching in mineral fertilizers

Similar to the above equation, phosphorous leaching from mineral fertilizers is calculated as phosphorous leaching from crop cultivation and current livestock number less the modeled livestock number.

```
e_OsparDuengerP(p)..

        sum((eg,inte,ek), 0.001 * p_OspDueP(ek,inte,eg)
        * v_Flaechen(p,ek,inte,eg))

        + 0.001 * sum(tk, p_Pabzug(tk) * p_AktBestand(tk))

        - 0.001 * sum(tk, p_Pabzug(tk) * v_Tierbestaende(p,tk))

        =E=

        v_OsparDueP(p) ;
```

Equation e_OsparFixierungN: Fixed nitrogen leaching

The model also includes the fixed nitrogen leaching levels from crop cultivation.

```
e_OsparFixierungN(p)..

        0.001 * sum((eg,inte,ek), p_OspFixN(ek,inte,eg)
        * v_Flaechen(p,ek,inte,eg))

        =E=

        v_OsparFixN(p) ;
```

Equation e_OsparDepositionN: Deposition of nitrogen leaching

The deposition of nitrogen leaching includes nitrogen leaching deposited from nitrogen leached from crop cultivation.

```
e_OsparDepositionN(p)..

        0.001 * sum((eg,inte,ek), p_OspDepN(ek) * v_Flaechen(p,ek,inte,eg))

        =E=

        v_OsparDepN(p) ;
```

Equation e_OsparDepositionP: Deposition of phosphorous leaching

We also calculate the deposition of phosphorous leaching, which is deposited from phosphorous leached from crop cultivation.

```
e_OsparDepositionP(p)..

        0.001 * sum((eg,inte,ek), p_OspDepP(ek) * v_Flaechen(p,ek,inte,eg))

        =E=

        v_OsparDepP(p) ;
```

Equation e_OsparEntzugPflanzeN: Nitrogen in plant-based food products

The nitrogen leaching quantity of plant-based food products includes nitrogen leaching from the supply of all domestically produced plant-based food products.

```
e_OsparEntzugPflanzeN(p)..

        0.001 * sum(nm, p_OspPflanzeN(nm) * v_AngebotNM(p,nm))

        =E=

        v_OsparPflanzeN(p) ;
```

Equation e_OsparEntzugPflanzeP: Phosphorous in plant-based food products

Similar to the above equation, phosphorous leaching of plant-based food products is calculated for the supply of summed domestically produced plant-based food products.

```
e_OsparEntzugPflanzeP(p)..

        0.001 * sum(nm, p_OspPflanzeP(nm) * v_AngebotNM(p,nm))

        =E=

        v_OsparPflanzeP(p) ;
```

Equation e_OsparEntzugTierN: Nitrogen in animal-based food products

Nitrogen leaching of animal-based food products includes nitrogen leaching from the supply of the total domestically produced animal-based food products.

```
e_OsparEntzugTierN(p)..

        0.001 * sum(nm, p_OspTierN(nm) * v_AngebotNM(p,nm))

        =E=

        v_OsparTierN(p) ;
```

Equation e_OsparEntzugTierP: Phosphorous in animal-based food products

Similarly, phosphorous leaching of animal-based food products is calculated as phosphorous leaching from the supply of total domestically produced animal-based food products.

```
e_OsparEntzugTierP(p)..

         0.001 * sum(nm, p_OspTierP(nm) * v_AngebotNM(p,nm))

         =E=

         v_OsparTierP(p) ;
```

Equation e_OsparTotalInputN: Total nitrogen leaching from agricultural inputs

Total nitrogen leaching from agricultural production inputs include nitrogen leaching from imported fodder, mineral fertilizers, as well as fixed and deposited nitrogen. The coefficients 2674 and 301 are nitrogen leaching values from nonmodeled activities and remain fixed.

```
e_OsparTotalInputN(p)..

         v_OsparFuN(p) + v_OsparDueN(p) + 2674 + 301 + v_OsparFixN(p)
         + v_OsparDepN(p)

         =E=

         v_OsparInputN(p) ;
```

Equation e_OsparTotalInputP: Total phosphorous leaching from agricultural inputs

Similarly, total phosphorous leaching from agricultural production inputs includes phosphorous leaching from imported fodder, mineral fertilizers, as well as fixed and deposited phosphorous. Coefficients 861 and 49 are phosphorous leaching values from nonmodeled activities and remain fixed.

```
e_OsparTotalInputP(p)..

         v_OsparFuP(p) + v_OsparDueP(p) + 861 + 49 + v_OsparDepP(p)

         =E=

         v_OsparInputP(p) ;
```

Equation e_OsparTotalOutputN: Total nitrogen leaching from food products

The total nitrogen leaching from food products consists of the sum of nitrogen leaching of plant- and animal-based food products that are calculated in previous equations *e_OsparEntzugPflanzeN* and *e_OsparEntzugTierN*, respectively.

```
e_OsparTotalOutputN(p)..

         v_OsparPflanzeN(p) + v_OsparTierN(p)

         =E=

         v_OsparOutputN(p) ;
```

Equation e_OsparTotalOutputP: Total phosphorous from food products

Phosphorous leaching from food products include phosphorous leaching from plant- and animal-based food products, which are calculated in previous equations *e_OsparEntzugPflanzeP* and *e_OsparEntzugTierP*, respectively.

```
e_OsparTotalOutputP(p)..

         v_OsparPflanzeP(p) + v_OsparTierP(p)

         =E=

         v_OsparOutputP(p) ;
```

Equation e_Biodiversitaet: Biodiversity area

In certain scenarios, we analyze the impacts of biodiversity policy goals on the food system. We model such scenarios by including a constraint on the biodiversity area. Here, we assume a constraint that 16.6% of agricultural area should be allocated for land uses that meet biodiversity criteria. We assume that land uses for biodiversity are natural extensive meadows, extensive grazing, fallow land with flowers, and rotational fallow. The area of these four land uses already occupy approximately 14.7% of agricultural area. Therefore, an additional 1.9% of agricultural area is allocated to biodiversity area over certain periods of time. This equation can be adjusted or (de)activated depending on the model scenario settings.

```
e_Biodiversitaet(p,eg)..

         sum(inte, v_Flaechen(p,"Naturwiesen extensiv",inte,eg)

         + v_Flaechen(p,"Weiden extensiv",inte,eg)

         + v_Flaechen(p,"Buntbrache",inte,eg)

         + v_Flaechen(p,"Rotationsbrache",inte,eg)
         )

         =E=

         (0.147 + 0.019 * p_KorrekturfaktorP(p))
         * v_FlaechenKG(p,eg, "KG Landwirtschaftliche Nutzfläche") ;
```

4.9 Income from agricultural production

The income module calculates the cash flow in the model, including net revenues, revenues, amount of direct payments, and export costs. At this stage of the modeling, we included only net revenues from agricultural production. Due to insufficient data, we were not able to calculate revenues and costs in the processing and trade stages as well as consumer expenditures. Depending on the objective function of the model, agricultural net revenues can be calculated in the model as an outcome of the model. Alternatively, agricultural net revenues can be used as an objective function and optimized, for example, to achieve the maximum possible agricultural net revenues.

Equation e_ExportPreis1: Export costs

The model includes the endogenous costs of exporting commodities. However, the current version of the model lacks information on the export costs and are not derived through the trade module considering the international and domestic markets with the supply and demand of commodities. We assume low and high boundaries for export costs. We use such cost calculations because the export volumes might become unrealistically large to have the high net revenues from higher prices of Swiss products than those in foreign countries. This assumption is particularly relevant in the objective function, where the agricultural net revenues are maximized.

In this equation, we consider the lower and upper levels of export costs, the sum of which is assumed to be equal to the export amount.

```
e_ExportPreis1(p,hp)..

          v_Exporte(p,hp)

          =E=

          (v_ExporteHochpreis(p,hp) + v_ExporteTiefpreis(p,hp))
          $ (p_Produktexporte(p,hp)>0) ;
```

Equation e_ExportPreis2: High export costs

Here, we constrain the high value of export costs. The high export costs are assumed to be lower than the export volume. Similar to the previous equation and subsequent equations where export costs are derived, we use this equation to derive export costs and restrict excessively high export volumes.

```
e_ExportPreis2(p,hp)..

          v_ExporteHochpreis(p,hp)

          =L=

          p_Produktexporte(p,hp) ;
```

Equation e_ExporteTiefpreisAll: Total low export costs

The low value of export costs is summed across all products with low export costs. These values are then used to derive the low export costs in the next equation.

```
e_ExporteTiefpreisAll(p)..

          sum(hp, v_ExporteTiefpreis(p,hp))

          =E=

          v_ExporteTiefpreisAll(p) ;
```

Equation e_ExporteTiefpreisCons: Low export costs

We assume that the low export costs of each commodity are equal to 10% of total low export costs.

```
e_ExporteTiefpreisCons(p,hp)..

          v_ExporteTiefpreis(p,hp)

          =E=

          0.1 * v_ExporteTiefpreisAll(p) ;
```

Equation e_ExporteTiefpreisMax: Maximum low export costs

The low export cost should be lower than five times the volume of export quantity.

```
e_ExporteTiefpreisMax(p,hp)..

         v_ExporteTiefpreis(p,hp)

         =L=

         5 * p_Produktexporte(p,hp) ;
```

Equation e_Exportdaten: Export costs

The mean export costs used in the model are calculated as the current export price less current import price of commodities, and their difference is multiplied by low export costs.

```
e_Exportdaten(p,hp)..

         0.001 * (p_Exportpreis(hp) - p_Importpreis(hp))
         * v_ExporteTiefpreis(p,hp)

         =E=

         v_Exportreduktion(p,hp) ;
```

Equation e_ErzeugungTot: Revenues from agricultural production

Revenues from agricultural production include revenues from crop area and livestock number less the export costs. The export costs refer to the costs related to exports of agricultural products; thus, agricultural revenues are not too large as a result of the higher prices of Swiss products than products produced abroad.

```
e_ErzeugungTot(p)..

         sum((ek,inte,eg), 0.001 * p_ErzeugungEK(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         + sum(tk, 0.001 * p_ErzeugungTK(tk) * p_FaktorGVE(tk)
         * v_Tierbestaende(p,tk))

         - sum(hp, v_Exportreduktion(p,hp))

         =E=

         v_Erzeugung(p) ;
```

Equation e_TierNumGrow: Growth of livestock number

We calculate the change in the livestock number with respect to the current livestock number. The results of this equation contribute to calculating the decrease in fixed costs of livestock whose number reduces.

```
e_TierNumGrow(p,tk)..

         v_Tierbestaende(p,tk) - p_AktBestand(tk)

         =E=

         v_TierbestPos(p,tk) - v_TierbestNeg(p,tk) ;
```

Equation e_DirzahlAktuel: Current direct payments for agricultural production

Agricultural production in Switzerland is subsidized via direct payments. Current direct payments for agricultural production include government payments for crop area and livestock number, which differ depending on crop type and livestock category. Here, we calculate the total amount of direct payments given for both crop cultivation area and number of livestock.

```
e_DirzahlAktuel..

*         Summe aktuelle Direktzahlungen
          v_AktuelDirzahl

          =E=

*         aktuelle Pflanzen Direktzahlen
          sum((ek,inte,eg), p_DirzahlEK(ek,inte,eg)
          * p_AktFlaeche(ek,inte,eg))

*         aktuelle Tier Direktzahlen
        + sum(tk, p_DirzahlTK(tk)
          * p_AktBestand(tk)) ;
```

Equation e_DirzahlEinsch: Maximum amount of direct payments for agricultural production

Direct payments for agricultural production cannot become too large. We assume that the total amount of direct payments given for agricultural production cannot be larger than their actual total amount.

```
e_DirzahlEinsch(p)..

*         Summe aktuelle Direktzahlungen
          v_AktuelDirzahl

          =G=

*         Pflanzen Direktzahlen
          sum((ek,inte,eg), p_DirzahlEK(ek,inte,eg)
          * v_Flaechen(p,ek,inte,eg))

*         Tier Direktzahlen
          + sum(tk, p_DirzahlTK(tk) * v_Tierbestaende(p,tk)) ;
```

Equation e_EinkommenTot: Total agricultural net revenues

Agricultural net revenues include revenues less costs from crop cultivation and livestock management. Agricultural revenues include flow of revenues per crop cultivation area and livestock unit number, and direct payments per crop area and livestock unit number. Costs consist of fixed and variable agricultural production costs as well as transportation costs of imported commodities and export costs. Fixed costs related to reduced livestock if the livestock number is lower than that in the actual observed situation. The reduction of these fixed costs is calculated for every period. The costs of imported products include transportation costs of the imports that reduce the monetary value of imported products. Export costs are the costs of exporting commodities, and these costs are needed to reduce very large export volumes.

Our agricultural net revenues include fixed revenue and cost values per hectare. In addition, we do not discount agricultural net revenues in our dynamic model because we want to observe the impacts of specific scenarios over time.

```
e_EinkommenTot(p)..
*        Pflanzen Einkommen
         sum((ek,inte,eg), 0.001 * p_ErzeugungEK(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

*        Pflanzen Direktzahlen
         + sum((ek,inte,eg), 0.001 * p_DirzahlEK(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

*        Pflanzen Variablekosten und Fixkosten
         - sum((ek,inte,eg), 0.001 * p_KostenVarEK(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         - sum((ek,inte,eg), 0.001 * p_KostenFixEK(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

*        Tier Einkommen
         + sum(tk, 0.001 * p_FaktorGVE(tk) * p_ErzeugungTK(tk)
         * v_Tierbestaende(p,tk))

*        Tier Direktzahlen
         + sum(tk, 0.001 * p_FaktorGVE(tk) * p_DirzahlTK(tk)
         * v_Tierbestaende(p,tk))

*        Tier Variablekosten und Fixkosten
         - sum(tk, 0.001 * p_FaktorGVE(tk) * p_KostenVarTK(tk)
         * v_Tierbestaende(p,tk))

*        Fixkosten für die Reduzierung des Tierbestaende
         - 0.001 * (1 - p_DiskKostFixTK(p)) * sum(tk, p_KostenFixTK(tk)
         * p_FaktorGVE(tk) * v_TierbestNeg(p,tk))

*        Fixkosten für steigende und gleichbleibende des Tierbestaende
         - 0.001 * sum(tk, p_KostenFixTK(tk) * p_FaktorGVE(tk)
         * v_Tierbestaende(p,tk))

         - 0.000001 * sum(hp, p_ImpTransKost(hp) * v_ImporteTot(p,hp))

         - sum(hp, v_Exportreduktion(p,hp))

         =E=

         v_Einkommen(p) ;
```

4.10 Employment

The model calculates employment to analyze the impacts of change in the food system on the employment in the system. We currently include only agricultural employment. Employment in the processing and trade sectors as well as changes in consumers' employment and in other sectors of the food system are not included due to insufficient data.

Equation e_FamilienarbeitstageTot: Family labor use for agricultural production

We calculate employment in the agricultural sector that includes family working days spent on crop production and livestock rearing. A change in agricultural employment is directly linked to the change in crop area and livestock unit number. We also assume an annual labor productivity increase of 1.2% based on discussions with experts.

```
e_FamilienarbeitstageTot(p)..

         p_LaborProd(p) *

         ( 0.001 * sum((ek,inte,eg), p_TageFamilieEK(ek,inte,eg)
         * v_Flaechen(p,ek,inte,eg))

         + 0.001 * sum(tk, p_TageFamilieTK(tk) * p_FaktorGVE(tk)
         * v_Tierbestaende(p,tk)) )

         =E=

         v_Familienarbeitstage(p) ;
```

5 Objective functions

In this section, we present the description of different objective functions. These objective functions can be extended and changed to minimization or maximization objectives depending on scenario settings. All the presented objective functions are used in the dynamic programming model to identify the optimal situations for the food system over the period of analyses. The programming approach can also be changed to a dynamic recursive programming approach.

Objective function e_ZielAngebotNM: Maximization of calorie supply

This function calculates the highest domestic supply of food products in terms of calories over the entire period of analysis. This objective function enables the identification of optimal amount of production, processing, trade, storage, and consumption of food products that result in the highest food calorie supply.

```
e_ZielAngebotNM..
         v_ZielAngebotNM
         =E=
         sum((p,nm), p_NM_Kaloriengehalt(nm) * v_AngebotNM(p,nm))  ;
```

Objective function e_ZielLandEinkommen: Maximization of agricultural net revenues

Here, we maximize agricultural net revenues over the entire modeled period. This objective function aims to find the highest net revenues of profitable agricultural production activities considering the settings of the food system. We do not discount net revenues to be able to disentangle the impacts of scenarios.

```
e_ZielLandEinkommen..
         v_ZielLandEinkommen
         =E=
         sum(p, v_Einkommen(p) )  ;
```

Objective function e_ZielMinRef: Reference scenario with minimization of deviations

This objective function is one of the reference scenarios in which the objective function minimizes the deviations of the variables of interest from the observed situation. With this objective function, we attempt to minimize the difference of modelled variables from the actual data.

To derive the variable *v_MinDev* in the objective function, we also model equations *e_ZielMinDev*, *e_FlaechenDev*, *e_TierbestaendeDev*, *e_NMRationenDev*, *e_ExporteDev, and e_ImporteDev* (see description of next equations).

```
e_ZielReferenz..
        v_ZielMinRef
         =E=
         sum(p, v_MinDev(p) ) ;
```

Objective function e_ZielMinDev: Minimization of deviations

Similar to the reference objective function, this objective function minimizes deviations of the variables of interest from the actual situation. This objective function is used to analyze the impacts of political, environmental, and technological change scenarios on the food system. We minimize the difference of modelled variables from the actual data. For this objective function, we also model equations *e_ZielMinDev, e_FlaechenDev, e_TierbestaendeDev, e_NMRationenDev, e_ExporteDev, and e_ImporteDev*.

```
e_ZielMinDev..

        v_ZielMinDev

         =E=

         sum(p, v_MinDev(p) ) ;
```

Equation e_MinDev: Sum of deviations

Equation *e_MinDev* includes the sum of deviation of variables from the actual situation—that is, deviations from the input data. The variables whose deviations are considered are crop area, livestock number, food consumption, export and import of commodities. The variable *v_ImportFM* is included to control for the high levels of concentrated feed import. The variables *v_FlaechenDev1, v_TierbestaendeDev1*, *v_NMRationenDev1*, *v_ExporteDev1*, and *v_ImporteDev1* include values that become lower than the actual situation. The variables *v_FlaechenDev, v_TierbestaendeDev, v_NMRationenDev, v_ExporteDev, and v_ImporteDev* include values that become larger than the actual situation. All these variables are summed up, and this summed value is minimized in the objective functions *e_ZielMinRef* or *e_ZielMinDev*.

```
e_MinDev(p)..

         v_MinDev(p)

         =E=

          sum((ek,inte,eg), v_FlaechenDev(p,ek,inte,eg)
         + v_FlaechenDev1(p,ek,inte,eg))

         + sum(tk, v_TierbestaendeDev(p,tk) + v_TierbestaendeDev1(p,tk))

         + sum(nm, v_NMRationenDev(p,nm) + v_NMRationenDev1(p,nm))

         + v_ImportFM(p)

         + sum(hp, v_ExporteDev(p,hp) + v_ExporteDev1(p,hp))

         +  sum(hp, v_ImporteDev(p,hp) + v_ImporteDev1(p,hp)) ;
```

Equation e_FlaechenDev: Deviation of crop area

Equation *e_FlaechenDev* calculates the relative deviation of modelled crop area variable from the actual crop area. The calculation is performed for each crop and its production method and regions cultivated.

```
e_FlaechenDev(p,ek,inte,eg) $ (p_AktFlaeche(ek,inte,eg)>0)..

           (v_Flaechen(p,ek,inte,eg) - p_AktFlaeche(ek,inte,eg))
          / p_AktFlaeche(ek,inte,eg)

         =E=

         v_FlaechenDev(p,ek,inte,eg) - v_FlaechenDev1(p,ek,inte,eg) ;
```

Equation e_TierbestaendeDev: Deviation of livestock number

Equation *e_TierbestaendeDev* calculates the relative deviation of modelled livestock category number from the observed livestock category number.

```
e_TierbestaendeDev(p,tk) $ (p_AktBestand(tk)>0)..

         p_FaktorGVE(tk) * (
            (v_Tierbestaende(p,tk) - p_AktBestand(tk))
          / p_AktBestand(tk) )

         =E=

          v_TierbestaendeDev(p,tk) - v_TierbestaendeDev1(p,tk) ;
```

Equation e_NMRationenDev: Deviation of food consumption

In this equation, we calculate the relative deviation of the modeled food consumption variable from the observed food consumption. Relative deviation is calculated for each food product.

```
e_NMRationenDev(p,nm) $ (p_AktuelleRation(nm)>0)..

           (v_NMRationen(p,nm) - p_AktuelleRation(nm))
          / p_AktuelleRation(nm)

         =E=

         v_NMRationenDev(p,nm) - v_NMRationenDev1(p,nm) ;
```

Equation e_ExporteDev: Deviation of exports

This equation calculates the relative deviation of modelled export volume variable from the observed export volume. For each exported commodity, we calculate the deviations of export volumes.

```
e_ExporteDev(p,hp) $ (p_Produktexporte(p,hp)>0)..

          (v_Exporte(p,hp) - p_Produktexporte(p,hp))
         / p_Produktexporte(p,hp)

         =E=

         v_ExporteDev(p,hp) - v_ExporteDev1(p,hp) ;
```

Equation e_ImporteDev: Deviation of imports

Equation *e_ImporteDev* calculates the relative deviation of modeled import volume variables from the actual import volume.

```
e_ImporteDev(p,hp) $ (p_Produktimporte(p,hp)>0)..

          (v_ImporteTot(p,hp) - p_Produktimporte(p,hp))
         / p_Produktimporte(p,hp)

         =E=

         v_ImporteDev(p,hp) - v_ImporteDev1(p,hp) ;
```

Objective function e_Umweltziel: Minimization of environmental impacts

This objective function minimizes the environmental impacts approximated by the ReCiPe single-score indicator. We minimize the ReCiPe from the production of raw crop and livestock products, processed products, by-products, replaced livestock by the next generation livestock, imported commodities and farm manure, and less frequently exported commodities.

This objective function was used when transferring the Green DSS-ESSA model codes written in LPL software into the SWISSfoodSys model written in GAMS software codes. Hence, this objective function was used in previous studies that employed the Green DSS-ESSA model (e.g., von Ow et al., 2020). See section 7 on the validation of programming code transfer from LPL software into GAMS.

```
e_Umweltziel..

        v_Umweltziel

         =E=

         sum(p,
           v_Umwelt_PRP(p,"Recipe")
         + v_Umwelt_TRP(p,"Recipe")
         + v_Umwelt_REM(p,"Recipe")
         + v_Umwelt_VRB(p,"Recipe")
         + v_Umwelt_KPD(p,"Recipe")
         + v_Umwelt_IMP(p,"Recipe")
         - v_Umwelt_EXP(p,"Recipe")
         + v_Umwelt_HFD(p,"Recipe")

         - v_Umwelt_HFD_N(p,"Recipe")
         ) ;
```

6 Running the model

The model runs in GAMS software and in file *Model.gms*. At the beginning of *Model.gms*, we refer to all other GAMS files—that is, *Sets.gms, Parameters.gms, Variables.gms* and *Equations.gms*, using the command *$include* (Figure 10). Using this command, *Model.gms* file calls out files with sets, parameters, variables, and equations. Any changes in definitions of sets, parameters, variables, and equations need to be specified in *Sets.gms, Parameters.gms, Variables.gms* and *Equations.gms*, respectively.

The model runs using the CPLEX solver suitable for the linear programming models. The commands *Iterlim, Reslim, Limrow*, and *Limcol* are used to reduce the number of model iterations, time running, and number of rows in the equation listing and number of columns listed in the variables listing. These commands can be deactivated.

```
$include Sets
$include Parameters
$include Variables
$include Equations

OPTION LP = cplex ;
OPTION ITERLIM = 10000000 ;
OPTION RESLIM  = 10000000 ;
OPTION LIMROW  = 0        ;
OPTION LIMCOL  = 0        ;
```

Figure 10: Specification of files with sets, parameters, variables, and equations in the model file.

After importing information from the external GAMS files and specifying the solver in *Model.gms*, a mathematical relationship of the model's equations in GAMS are given—that is, objective function and constraints. All the equations that are modelled are listed after their mathematical relationships are specified. Thereafter, the command solving the optimization process is given. Figure 11 presents an example of coding the list of modeled variables and commands used to solve the model.

```
Model AngebotModell
/
* Zielfunktion
e_ZielAngebotNM


* Pflanzenbau-Modell
e_pa
e_pb
e_p1b
e_pc8b
e_pc8c
e_pd
/
;

Solve AngebotModell maximizing v_ZielAngebotNM using lp ;
```

Figure 11: List of equations modelled and command to implement optimization process.

7 Validation of the model transfer

In this section, we present the validation of the model transferred from LPL software to GAMS software. The validation of the transferred model is done by comparing the results of the Green DSS-ESSA model simulated in GAMS and LPL. In the Green DSS-ESSA model, the objective function minimizes environmental impacts of ReCiPe from agricultural production, processing, import, and export. ReCiPe is a life cycle impact assessment method that enables the aggregation of the three areas of protection human health, ecosystem services, and resource use into a single-score using a set of weights. ReCiPe outputs from agricultural production, farm manure, processing, and import are added to the objective value, while ReCiPe outputs from exports and farm manure are subtracted. The model is dynamic and simulated for the period 2014–2020.

For simplicity of the interpretation of results, we present the results of the main variables, including objective value, environmental impacts, crop area, and livestock number. Tables 6, 7, and 8 present the relative difference between GAMS and LPL in terms of the results on objective value and environmental impacts, crop area, and livestock number, respectively.

The validation of the model transfer reveals that the objective function value of the SWISSfoodSys model simulated in GAMS software does not differ much from the objective value of the Green DSS-ESSA model simulated in LPL software. The results of the objective value simulated in GAMS differs from LPL by 0.045% (Table 6). However, the results of certain elements of the variables substantially differ between GAMS and LPL. For example, the difference between GAMS and LPL in ReCiPe outputs from livestock replacement by next generation substantially differ each year, where the difference is approximately 85% in the first simulated year (i.e., 2014). However, the sum of the environmental impacts from variables do not differ substantially. For example, the difference between GAMS and LPL of the summed over years of ReCiPe outputs livestock replacement by next generation is 0.9%.

We also observed large differences for certain elements of variables between the simulated results of GAMS and LPL. These large differences can be due to the use of different software types and solvers.

Table 6: Comparison of GAMS and LPL results on objective value and environmental impacts.

Variable	Years	GAMS results	LPL results	% difference between GAMS and LPL
Objective value	Sum over years	6,777,763	6,780,783	-0.045%
Environmental impacts from different modules				
ReCiPe outputs from crops	2014	544,193	544,295	-0.02%
	2015	546,170	544,447	0.32%
	2016	548,443	546,659	0.33%
	2017	550,451	548,119	0.42%
	2018	552,461	549,986	0.45%
	2019	555,443	552,747	0.49%
	2020	556,553	552,485	0.73%
	Sum over years	3,853,715	3,838,737	0.39%
ReCiPe outputs from livestock	2014	347,804	347,482	0.09%
	2015	239,979	232,289	3.20%
	2016	222,474	221,937	0.24%
	2017	199,171	201,126	-0.98%
	2018	179,358	185,582	-3.47%
	2019	168,068	169,680	-0.96%
	2020	155,634	153,519	1.36%

Variable	Years	GAMS results	LPL results	% difference between GAMS and LPL
	Sum over years	1,512,488	1,511,616	0.06%
ReCiPe outputs from replacement of livestock for the next generation	2014	2,638	408	84.52%
	2015	17,982	17,982	0.00%
	2016	4,703	5,629	-19.69%
	2017	9,354	11,874	-26.94%
	2018	17,087	17,309	-1.30%
	2019	13,295	11,088	16.60%
	2020	7,463	7,546	-1.11%
	Sum over years	72,523	71,838	0.94%
ReCiPe outputs from processing	2014	100,336	99,559	0.77%
	2015	103,378	103,164	0.21%
	2016	89,345	88,671	0.76%
	2017	86,331	86,284	0.05%
	2018	84,841	85,931	-1.28%
	2019	84,480	84,133	0.41%
	2020	75,229	75,109	0.16%
	Sum over years	623,940	622,851	0.17%
ReCiPe outputs from processing of by-products	2014	10,076	9,720	3.54%
	2015	9,327	9,132	2.10%
	2016	7,763	7,444	4.10%
	2017	6,775	7,106	-4.89%
	2018	6,794	7,259	-6.85%
	2019	7,147	7,195	-0.67%
	2020	6,372	6,246	1.97%
	Sum over years	54,253	54,102	0.28%
ReCiPe outputs from importing commodities	2014	384,595	436,412	-13.47%
	2015	391,516	412,803	-5.44%
	2016	355,890	281,094	21.02%
	2017	325,376	359,602	-10.52%
	2018	327,453	313,258	4.34%
	2019	338,789	341,975	-0.94%
	2020	349,151	339,333	2.81%
	Sum over years	2,472,770	2,484,478	-0.47%
ReCiPe outputs from exporting commodities	2014	257,096	255,889	0.47%
	2015	257,058	255,854	0.47%
	2016	257,058	255,798	0.49%
	2017	257,058	255,798	0.49%

Variable	Years	GAMS results	LPL results	% difference between GAMS and LPL
	2018	257,103	255,842	0.49%
	2019	257,103	255,842	0.49%
	2020	257,103	255,842	0.49%
	Sum over years	1,799,579	1,790,865	0.48%
ReCiPe outputs from farm manure	2014	3,900	3,655	6.28%
	2015	617	536	13.05%
	2016	214	245	-14.92%
	2017	173	257	-48.30%
	2018	880	972	-10.46%
	2019	2,375	2,305	2.91%
	2020	4,189	4,001	4.49%
	Sum over years	12,348	11,973	3.04%

If we examine the simulation results of crop areas in GAMS and LPL, then, in most cases, both software types produce the same results over the years (Table 7). Only the areas of certain crops differ in certain years and in certain regions. Broad bean, fodder beet, and winter triticale have the largest differences in crop areas between the software types, but the differences in these crops are only present for some years.

Table 7: Comparison of GAMS and LPL results on crop area—percentage difference.

Years	Region	Winter barley	Summer barley	Summer oats	Winter triticale	Winter wheat	Summer wheat	Winter rye	Winter grain	Grain maize	Seed maize	Silo maize	Sugar beet	Fodder beet	Pota-toes	Rape	Sun-flower
2014	Valley	19	0	0	0	1	0	0	0	-23	0	0	0	0	0	-2	19
2014	Hill	0	0	0	0	1	0	0	6	0	0	0	0	0	0	0	0
2014	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2015	Valley	-18	0	0	0	1	0	0	0	9	0	0	0	9	16	-2	-18
2015	Hill	0	0	0	0	1	0	0	6	0	0	0	0	0	0	0	0
2015	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2016	Valley	-14	0	0	0	1	0	-14	0	11	0	0	0	9	4	5	-14
2016	Hill	0	0	0	7	-2	0	0	6	0	0	0	0	43	0	0	0
2016	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017	Valley	-7	0	0	-40	-3	0	10	-23	11	0	0	0	18	11	3	-7
2017	Hill	0	0	-25	13	-3	0	0	6	0	0	0	0	30	0	0	0
2017	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2018	Valley	38	0	0	-15	-31	0	14	-23	11	0	0	0	18	21	2	38
2018	Hill	0	0	-25	19	-3	0	13	6	0	0	0	0	0	0	0	0
2018	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2019	Valley	0	0	0	0	17	0	-24	-23	11	0	0	0	0	-35	-15	0
2019	Hill	0	0	-25	0	2	0	0	6	0	0	0	0	0	0	0	0
2019	Mountain	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2020	Valley	0	0	0	0	7	0	-24	-23	11	0	0	0	0	-35	23	0
2020	Hill	0	0	-25	0	3	0	0	6	0	0	0	0	0	0	0	0
2020	Mountain	5	40	0	0	0	43	43	43	0	0	0	0	0	0	0	5
Averaged over years		1	7	-5	0	-2	3	-5	1	1	0	0	0	7	-3	1	-4

Table 7: Comparison of GAMS and LPL results on crop area—percentage difference (continued).

Years	Region	Soy	Broad bean	Pea	Seed potato	Field vege-tables	Legu-mes	Field can vege-tables	Annual rene-wable raw materials	Tobacco and other annual crops	Set-aside arable land	Cereals	Other arable crops	Vegetable, miscel. and set-aside arable land	Open and set-aside arable land	Artificial meadow	Pome fruit
2014	Valley	0	78	0	0	25	0	0	0	0	0	-2	1	2	0	0	0
2014	Hill	0	0	-45	0	0	0	0	0	0	0	0	-1	0	0	0	0
2014	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2015	Valley	0	78	0	0	25	0	0	0	0	0	-4	6	2	0	0	0
2015	Hill	0	0	-45	0	0	0	0	0	0	0	0	-1	0	0	0	0
2015	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2016	Valley	0	78	0	0	25	0	0	0	0	0	-3	3	3	0	0	0
2016	Hill	0	43	-40	0	0	0	0	0	0	0	0	1	0	0	0	0
2016	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017	Valley	0	0	0	0	25	0	0	0	0	0	-4	5	4	0	0	0
2017	Hill	0	73	-40	0	0	0	0	0	0	0	0	0	0	0	0	0
2017	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2018	Valley	0	0	0	0	25	0	0	0	0	0	-4	5	4	0	0	0
2018	Hill	0	0	-40	0	0	0	0	0	0	0	1	-2	0	0	0	0
2018	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2019	Valley	0	0	0	0	25	0	0	0	0	0	8	-15	5	0	0	0
2019	Hill	0	0	-40	0	0	0	0	0	0	0	1	-3	0	0	0	0
2019	Mountain	0	0	0	0	0	0	0	0	0	0	8	0	0	6	-9	0
2020	Valley	0	0	0	0	25	0	0	0	0	0	2	-10	7	0	0	0
2020	Hill	0	0	-40	0	0	0	0	0	0	0	1	-4	0	0	0	0
2020	Mountain	0	0	0	0	0	0	0	0	0	0	14	0	0	9	-9	0
Averaged over years		0	20	-23	0	24	0	0	0	0	0	-1	-1	4	0.04	0.1	0

Table 7: Comparison of GAMS and LPL results on crop area—percentage difference (continued).

Years	Region	Stone fruit	Berries	Other permanent crops	Arable	Permanent crop	Natural meadow	Natural meadow med-intensive	Natural meadow extensive	Vegetable in protect cultivation	Litter, peat, hedge, copse	Other crops in protect cultivation	Agric land	Summering pasture	Cultivate are
2014	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2014	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2014	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2015	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2015	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2015	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2016	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2016	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2016	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2017	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2018	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2018	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2018	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2019	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2019	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2019	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2020	Valley	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2020	Hill	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2020	Mountain	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Averaged over years		0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8 presents the relative (%) difference of GAMS results on livestock number in comparison to LPL. The table shows that the model simulated in GAMS, in most cases, generates the same number of livestock as the model simulated in LPL. However, the numbers of a few of the livestock categories substantially differ (in %) between the two models.

Table 8: Comparison of GAMS and LPL results of livestock number—percentage difference.

Livestock		**Years**						
	2014	**2015**	**2016**	**2017**	**2018**	**2019**	**2020**	**Averaged over years**
Cows for transport milk production	0	0	0	0	-22	0	-1	-2
Cows milked, no traffic milk production	0	0	0	0	0	0	0	0
Cattle over two years old	0	0	-30	-16	23	0	-1	-2
Cattle one to two years old	0	-31	-16	23	0	-1	0	-2
Bulls over two years old	0	0	0	0	0	29	47	18
Bulls one to two years old	0	0	0	0	75	59	18	35
Female cattle for breeding, 4–12 months old	0	-44	26	0	0	0	1	-1
Male cattle for breeding, 4–12 months old	0	0	0	0	102	23	1	30
Rearing calves under four months old, female	0	-48	59	0	-4	0	0	1
Rearing calves under four months old, male	0	0	0	0	112	0	0	24
Suckler and nurse cows (excluding calves)	0	0	0	0	4	2	2	1
Cattle from suckler and nurse cows, 1–2 years	0	0	0	0	0	1	2	1
Calves of suckler and nurse cows, under one year old	0	0	0	0	1	2	2	1
Cattle, bulls and steers (large cattle fattening) over four months	0	0	0	0	0	0	1	1
Calves for fattening under four months old	2	22	-31	-5	-11	0	0	-2
Calves for fattening	0	0	0	0	0	0	0	0
Sausage calves	2	0	0	0	0	0	0	2
Suckling and non-lactating breeding sows over six months old	0	0	0	0	0	0	0	0
Breeding boar	0	0	0	0	0	0	0	0
Replacement of livestock by next generation up to six months old and fattening pigs	0	0	0	0	0	0	0	0
Suckling pig	2	0	0	0	0	0	0	2
Lactating and pregnant mares	0	0	0	0	0	0	0	0
Foal at foot	1	1	1	1	1	1	1	1

Livestock		Years						
	2014	2015	2016	2017	2018	2019	2020	Averaged over years
Other horses over three years of age	0	0	0	0	0	0	0	0
Other foals under three years of age	0	0	0	0	0	0	0	0
Sheep for milking	0	0	0	0	0	0	0	0
Other female sheep over one year old	0	2	2	2	2	2	2	2
Ram over one year of age	0	-12	-16	-8	-4	-1	0	-4
Young ewe under one year of age (female and male)	3	1	2	2	2	2	2	2
Lamb for slaughter	-4	-1	-1	0	0	0	0	-1
Goat for milking	0	0	0	0	0	0	0	0
Other female goats over one year of age	0	0	0	0	0	0	0	0
Goat male over one year of age	0	0	0	0	0	0	0	0
Young goat under one year of age (female and male)	0	0	0	0	0	0	0	0
Slaughter goats	-2	-2	-2	-2	-2	-2	-2	-2
Breeding hen and rooster (laying and fattening lines)	0	0	2	0	-4	11	0	1
Laying hens	2	-2	2	2	-2	2	12	2
Pullets, young hens and chicks (without fattening chicken)	-3	0	1	2	-6	15	24	1
Chickens for fattening of any age	2	1	1	1	0	0	0	1
Cows with intensive method for traffic milk production	9	5	1	-6	3	3	3	2
Cows with extensive method for traffic milk production	0	0	0	0	0	0	0	0

8 References

Agristat, 2024. Statistische Erhebungen und Schätzungen über Landwirtschaft und Ernährung 2023. Kapitel 7: Nahrungsmittelbilanz. Schweizer Bauernverband, Brugg.

BRP and BLW, 1992. Sachplan Fruchtfolgeflächen (FFF), Festsetzung des Mindestumfanges der FFF und deren Aufteilung auf die Kantone. Bundesamt für Raumplanung (BRP) und Bundesamt für Landwirtschaft (BLW), Bern.

Drewnowski, A., Amanquah, D., Gavin-Smith, B., 2021. Perspective: How to Develop Nutrient Profiling Models Intended for Global Use: A Manual. Advanced Nutrition 12(3), 609–620.

Fulgoni, V. L., 3rd, Keast, D. R., andDrewnowski, A., 2009. Development and validation of the nutrient-rich foods index: a tool to measure nutritional quality of foods. Journal of Nutrition, 139(8), 1549–1554.

Global Burden of Disease, 2020. Global burden of 87 risk factors in 204 countries and territories, 1990-2019: A systematic analysis for the Global Burden of Disease Study 2019. Lancet 396(10258), 1223–1249.

Korkaric, M., Lehto, M., Poiger, T., de Baan, L., Mathis, M., Amman, L., Hanke, I., Balmer, M., Blom, J.F., 2023. Nationale Risikoindikatoren für Pflanzenschutzmittel, Weiterführende Analysen. Agroscope Science, 1–48.

Stylianou, K. S., Fulgoni, V. L., Jolliet, O., 2021. Small targeted dietary changes can yield substantial gains for human health and the environment. Nature Food, 2(8), 616–627.

von Ow, A., 2019. Ernährungssicherungssystem DSS-ESSA Decision Support System – Ernährungssicherungsstrategie Angebotslenkung: Dokumentation. In Bundesamt für Wirtschaftliche Landesversorgung (BWL) (p. 110).

von Ow, A., Waldvogel, T., Nemecek, T., 2020. Environmental optimization of the Swiss population’s diet using domestic production resources. Journal of Cleaner Production, 248, 119241.

Appendix

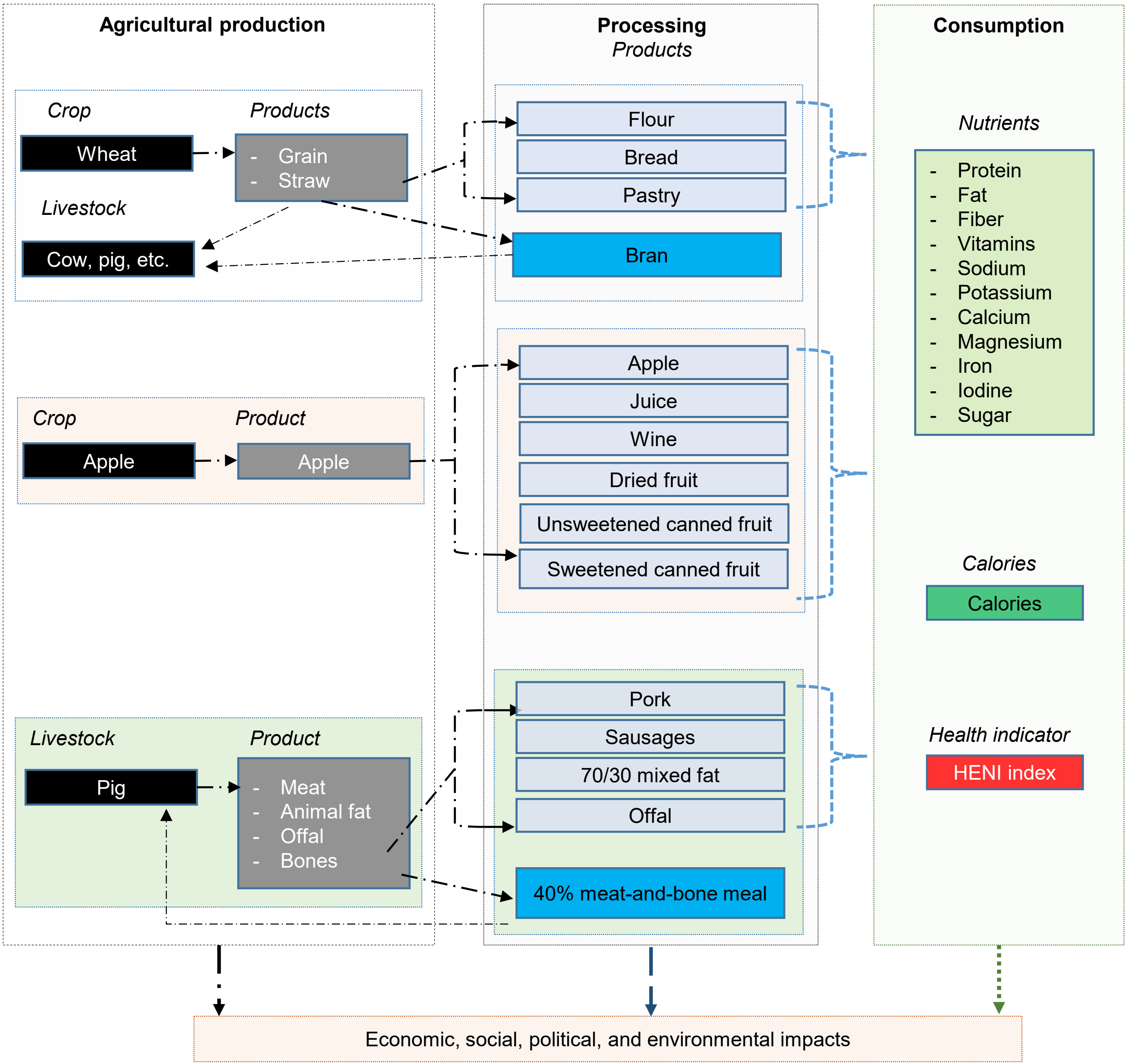


Figure 12: Example of flow of products of wheat, apple, and pig in agricultural production, processing, and consumption stages of the model.

9 List of Figures

10 List of Tables