

**MIND
STEP**



MODELLING INDIVIDUAL DECISIONS TO SUPPORT THE EUROPEAN POLICIES RELATED TO AGRICULTURE

Deliverable D7.2: Beta version of a wrapper to allow a standardized communication channel between the models

AUTHORS	Marc Müller (WR), Hugo Scherer (WR), Alexander Gocht (Thuenen)
APPROVED BY WP MANAGER:	Alexander Gocht (Thuenen)
DATE OF APPROVAL:	10.06.2022 Update version: 05.12.2022
APPROVED BY PROJECT LEADER:	John Helming (WR)
DATE OF APPROVAL:	Update version 07.12.2022
CALL H2020-RUR-2018-2	Rural Renaissance
WORK PROGRAMME Topic RUR-04-2018	Analytical tools and models to support policies related to agriculture and food - RIA Research and Innovation action
PROJECT WEB SITE:	https://mind-step.eu

This document was produced under the terms and conditions of Grant Agreement No. 817566 for the European Commission. It does not necessary reflect the view of the European Union and in no way anticipates the Commission's future policy in this area.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 817566.



TABLE OF CONTENTS

EXECUTIVE SUMMARY	2
1. INTRODUCTION.....	3
2. DESCRIPTION OF EXAMPLE TASKS.....	3
2.1. RISK MODELLING IN MIND STEP	4
2.2. TK UTILITY FUNCTION IN FARMDYN	5
3. ESTABLISHING A WORKFLOW	6
4. CONCLUSIONS.....	9
5. ACKNOWLEDGEMENTS	9
6. REFERENCES.....	9



ACRONYMS

AML	Algebraic Modelling Language
GAMS	General Algebraic Modelling System
GDX	GAMS Data Exchange Format
GGIG	GAMS Graphical Interface Generator
GUI	Graphical User Interface



EXECUTIVE SUMMARY

The MIND STEP toolbox is a suite of models intended to be a modular framework where functionality can be added with additional models and data and where the models can be executed in flexible combinations. The latter requires to embed the models in the toolbox following protocols developed in other work packages (namely WP2.2, WP3.1 and WP4.1) by using software wrappers which permit the execution of each model from an external environment.

This deliverable 7.2. describes a linkage between an empirical model concerning farmers' risk attitudes and the simulation model FarmDyn using wrapper functions which permit the creation of the required datasets and the execution of the FarmDyn model from an R environment. The R programming language was chosen as a majority of modelling team within the MIND STEP project use R in connection with their respective simulation models. The wrapper function to execute FarmDyn is therefore included in a downloadable R package, which contains also further functionalities, for instance permitting the aggregation of individual farm data to groups thereof or representative farms. Aggregating data, combining different sources of information, and executing the simulation model represents a typical workflow between the farm-level models developed in work package 3 of the MIND STEP project, which can be extended to other tasks and simulation models used within the project.



1. INTRODUCTION

The MIND STEP suite of models is intended to be a modular framework where functionality can be added with additional models and data and where the models can be executed in flexible combinations. The latter requires to embed the models in the toolbox following protocols developed in other work packages (namely WP2.2, WP3.1 and WP4.1) by using software wrappers which permit the execution of each model from an external environment. The wrapper concept reduces the work to adjust the models for the toolbox and ensures that arguments and data are standardized and well documented. While it would be preferable to design a generic wrapper, the manifold solutions foreseen in this project with respect to programming languages and the number of interlinkages, specific wrappers will be needed. A prototype wrapper function was implemented in the R programming language, as described in MIND STEP Deliverable 7.1. It permits the execution of a GAMS-based simulation model, exploiting some of the features of the applied user interface for model set-up and parallel execution, and the exchange of complex data structures between models. The R language facilitates handling of complex data structures and is an environment model developers within MIND STEP are familiar with, thus the wrapper function prototype can serve as a tool to facilitate the integration of the MIND STEP toolbox.

This report describes the application of the concept of wrapper functions to a specific case, namely the linkage between empirical estimates of farmers' attitudes towards risk and a farm-level simulation model.

2. DESCRIPTION OF AN EXAMPLE TASK

The research activities in MIND STEP at farm level comprise empirical studies on farmers' preferences for the adaptation of GHG mitigation options (Task 3.3), the adjustments of crop management systems in case of changing prices (Task 3.4), and the attitudes towards risk (Task 3.5). Integrating the results from these empirical tasks and using them to improve the empirical grounding of an individual decision making (IDM) simulation model at farm-level is the main objective of Task 3.2. At the time of writing this report, the research activities on risk attitudes were the most advanced and could be used as test case for establishing an example work flow of exchanges between models. Figure 1 depicts the honeycomb of models and tools as envisaged in the MIND STEP proposal. The IDM model FarmDyn covers already a wide range of topics relevant for this honeycomb, for instance animal feed rations, crop rotations, and cost accounting, as represented by the grey-framed hexagons. It also covers crop management, risk management, and GHG emissions; the MIND STEP tasks 3.3 to 3.5 aim at improving the empirical grounding of these parts of the model (red and blue hexagons in Figure 1).



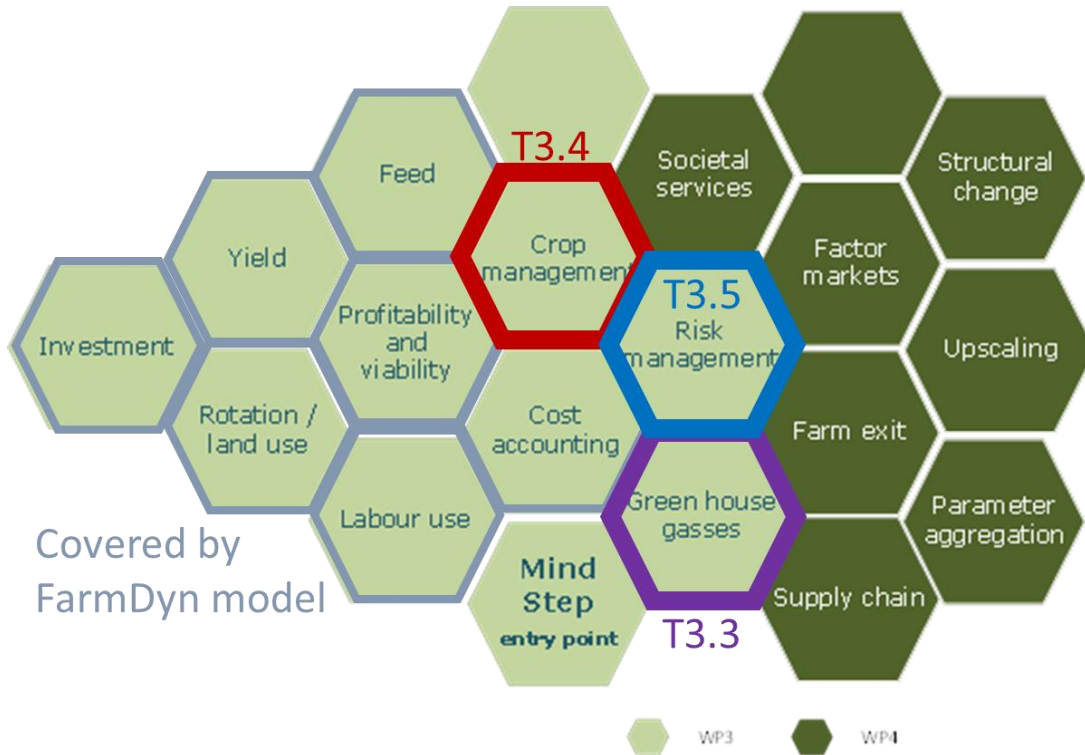


Figure 1 MIND STEP: system of interlinked models centred around individual decision making units

2.1. Risk Modelling in MIND STEP

The empirical studies on risk behaviour carried out in Task 3.5 are based on the concept of Cumulative Prospect Theory (CPT, Tversky and Kahnemann 1992, TK), which is more flexible in describing risk utility compared to expected utility and other approaches. It was found to describe more accurately risk behaviour in many empirical studies because it permits different individual weightings for possible outcomes of farming decisions, for instance by giving higher weights to potential shortfalls, which is plausible in the case of risk-averse farmers. The shape of the used utility function depends on a range of farm characteristics, for instance age and university degree of the farm manager as well as acreage and economic size. Figure 2 depicts an example for the empirical results for a sample of farmers in Germany.

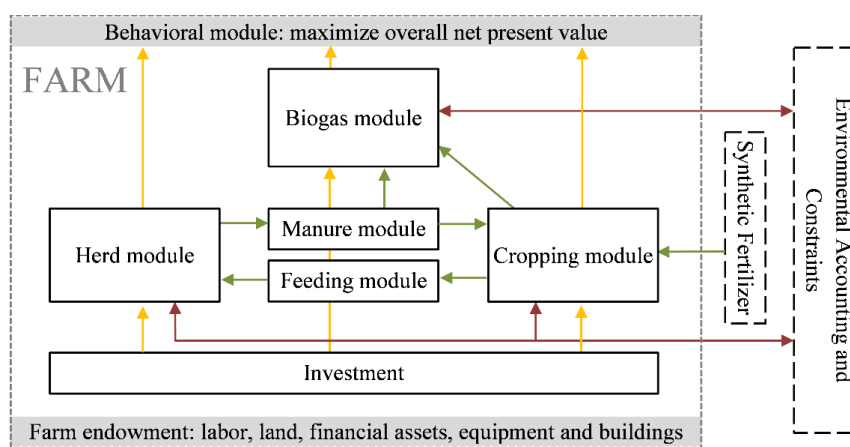
Variables	Description	Coefficients b to calculate pros		
		b_sigma	b_gamma	b_lambda
age	= age in years/10	-0.04	-0.05	0.52
sex	= 1 if male; else 0	-0.01	-0.11	-2.57
UniDegree	= 1 if university degree; else 0	0.08	0.07	-2.15
logArabLa	= LN(hectar cultivated arable land)	-0.01	0.00	0.96
FullTime	= 1 if full time; else 0	-0.05	-0.06	-1.27
PropSales	= Sales revenue from arable farming / total Sales revenue	0.02	0.02	-0.48
cons	Constant	0.69	0.93	1.22

Figure 2 Estimated coefficients for TK utility function parameters (Source: personal communication with MIND STEP partners at Thuenen Institute involved in WP3)

Based on these coefficient estimates, it is possible to derive TK utility function parameters for individual farms once the data for the shown farm characteristics are available.

2.2. TK Utility Function in FarmDyn

The single farm programming model FarmDyn (Britz et al. 2016) allows simulating optimal farm management and investment decision under changes in boundary conditions such as prices, technology or policy instruments. It is based on a model template for a fully dynamic or comparative-static bio-economic simulation building on Mixed-Integer Programming. Figure 3 provides an overview on the blocks of equations (modules) of FarmDyn. This modular set-up permits also the inclusions of different risk behavioural models as shown in the screenshot of the FarmDyn GUI in Figure 4.



Source: <https://www.ilr.uni-bonn.de/em/rsrch/farmdyn/FarmDynDoku/template/introduction/>

Figure 3 FarmDyn Template model

The risk utility function feature is a recent addition to the FarmDyn model developed in the context of the MIND STEP project and described in Britz (2022). It is based on a linear approximation of the TK utility function and the simulation of a range of likely outcomes of farming decisions and permits the assessment of risks related to crop yield and crop prices. The Shape of the TK utility function is determined by the parameters visible in the screenshot (Figure 4), for which default values are provided. Based on the estimated coefficients presented in (Figure 2), the risk utility function parameters can be updated. This update procedure is part of the workflow described in the following.

Risk behavior		
Risk model	Estimated utility function	
TK utility function alpha	None	0.88
TK utility function beta	Estimated utility function	0.88
TK utility function gamma	NPV at risk	-2.22
TK weighting function	MOTAD against NPV	0.65
	MOTAD against target	
	Target-Motad	
	Expected shortfall	
Points to estimate approximation		200
Segments on piece-wise approximation		10

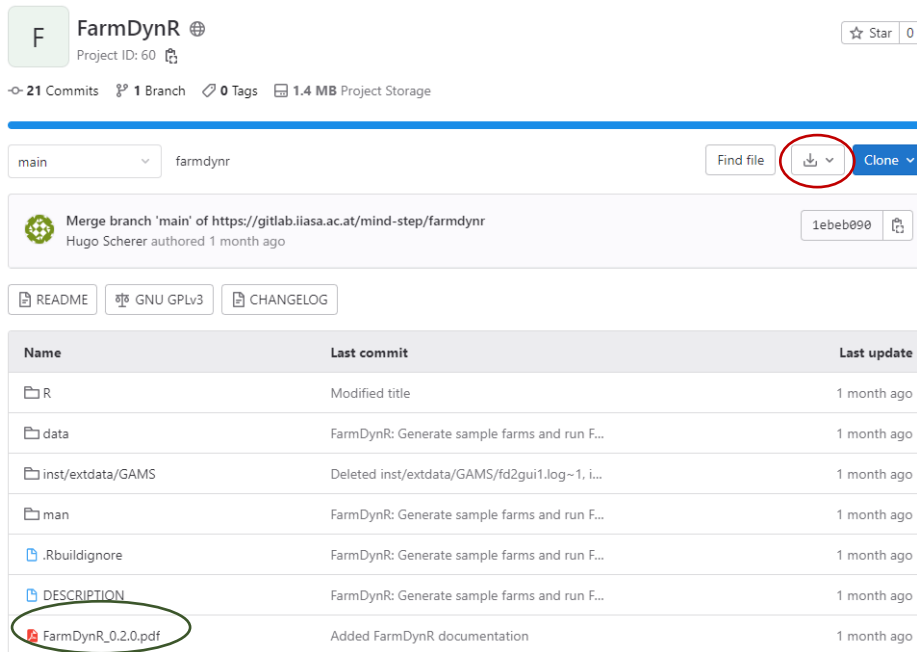
Figure 4 Risk behaviour options in FarmDyn

FarmDyn can be executed for individual farms in national or EU-wide FADN datasets or for specific groups (e.g. for average dairy farms in a region). Routines for the extraction of the datasets required by FarmDyn are available for standard settings and for some national as well as EU-wide FADN datasets. The computation of the utility function parameters requires an additional set of information like educational status of the farm manager and other farm characteristics not included in the default datasets. This has to be provided by the user in the form of a.gdx container named “farmCharacteristics.gdx” in Figure 6.

3. ESTABLISHING A WORKFLOW

Establishing a workflow using the wrapper function in the FarmDynR package requires first to have FarmDyn installed and a dataset for farm samples for which FarmDyn should be executed is available. Instructions on the installation of FarmDyn can be found at the documentation homepage maintained by the developer team at the University of Bonn: <https://www.ilr.uni-bonn.de/em/rsrch/farmdyn/FarmDynDoku/installation/>. A template for the structure and contents of the required farm sample file can be found in the FarmDyn “dat” directory in a.gdx-container named “farmData_DE.gdx”.

The next step to make use of the wrapper functions is to install the R package FarmDynR. The package can be found at the IIASA GitLab server in <https://gitlab.iiasa.ac.at/mind-step/farmdynr>. As shown in Figure 5, the package can be downloaded as a whole (red circle in Figure 5) or, if desired, the R-style documentation file can be examined before downloading (green circle).



FarmDynR Project ID: 60 ☆ Star 0

21 Commits 1 Branch 0 Tags 1.4 MB Project Storage

main farmdynr Find file ↓ Clone ↓

Merge branch 'main' of https://gitlab.iiasa.ac.at/mind-step/farmdynr
Hugo Scherer authored 1 month ago 1ebeb090

README GNU GPLv3 CHANGELOG

Name	Last commit	Last update
R	Modified title	1 month ago
data	FarmDynR: Generate sample farms and run F...	1 month ago
inst/extdata/GAMS	Deleted inst/extdata/GAMS/fd2gui1.log~1, i...	1 month ago
man	FarmDynR: Generate sample farms and run F...	1 month ago
.Rbuildignore	FarmDynR: Generate sample farms and run F...	1 month ago
DESCRIPTION	FarmDynR: Generate sample farms and run F...	1 month ago
FarmDynR_0.2.0.pdf	Added FarmDynR documentation	1 month ago

Figure 5 Screenshot of the FarmDynR site at the IIASA GitLab repository

As mentioned above, datasets for a sample of farms should be available in the required format. In some cases, it is deemed practical to aggregate the farm data from individual level to groups of farms or representative farm types. This can be done using the “updateFarmData” function, which requires the basic farm data to aggregate from as well as a mapping of individual farm IDs to the targeted groups. This information should be provided in GDX format. In R, the GDX file containing all the necessary information can be read with the “updateFarmData” function, which takes the GDX filename, the directory where it is located, the name of the set where the farm characteristics are linked to farm IDs, and the name of the characteristics (e. g. “Region”, “Conventional/Organic”, etc.). This function then makes sample farms based on the farm characteristics by using the weighted mean for farm data and the mode for FarmDyn code variables. It then creates a new GDX file with the pattern farmData_[mapping].gdx (optionally, this functionality can be switched off to inspect the data in R). In addition, descriptive statistics can be generated with the same inputs and a selection of variables by using the “groupstats” function.

By setting the optional argument “cptcoeffs” to “TRUE”, farm-specific parameters of the risk utility function are calculated. To do this, “updateFarmData” requires that the user provides the paths to the locations of farmCharacteristics and estimated coefficients. In the example shown in Figure 6, the coefficients are read from an excel file “utilityCoefficients.xlsx”.

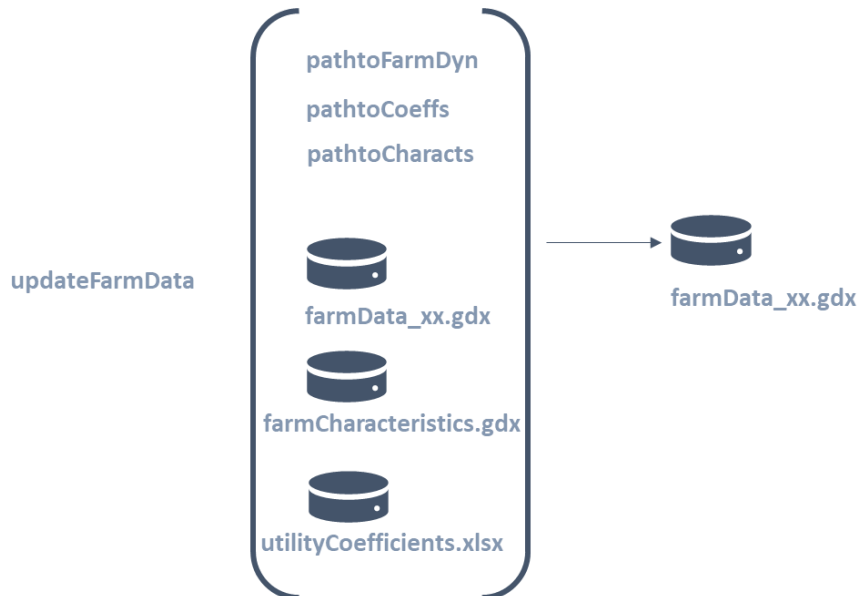


Figure 6 Schematic view of the wrapper “updateFarmData”

Connecting the empirically estimated utility function parameters and individual farm instances of FarmDyn requires updating the input data for parallel runs of a sample of farms. As described in Deliverable 7.1, for the execution of multiple instances of FarmDyn, the user has to provide the wrapper function “runFarmDynfromBatch” with the following information:

- Path to the FarmDyn directory
- Name of an xml file that governs the model-specific appearance of the GUI (e.g. GUIsettings.xml), available by default in the /GUI subdirectory of the FarmDyn directory
- Name of an ini-file that contains user-specific information like the GAMS version to be used (e.g. userSettings.ini), by default in the /GUI subdirectory of the FarmDyn directory
- Name and location of the scenario file used for the batch execution (scenarioBatch.txt) in a location chosen by the user

Creating the latter three files can easily be done by launching the GUI, providing the required settings, and executing the FarmDyn model once. Following the instructions in the GGIG user manual, a marked block of the automatically generated file “runInc.gms” in the /incgen subdirectory of the FarmDyn directory can be copied to the “scenarioBatch.txt” file. This can be done automatically from R with the help of the “writeBatch” function, which takes the information necessary from “runInc.gms”, adds modifications to the text, and saves it to a “.txt” file. This is less error-prone and quicker than doing it manually. Once these preparatory steps are taken, the model can be executed from an R environment by calling the function “runFarmDynfromBatch”. With the paths to the FarmDyn directory, the two files for the GUI setup (GUISettings.xml, userSettings.ini) and the file for the scenario specification (scenBatch.txt), the wrapper functions generates a temporary command-file containing all the relevant information to execute the model within the GUI. A system command to the GUIExecute.bat file is then issued and the model is run. Figure 7 gives a schematic overview on the steps and files involved in the use of “runFarmDynfromBatch”.

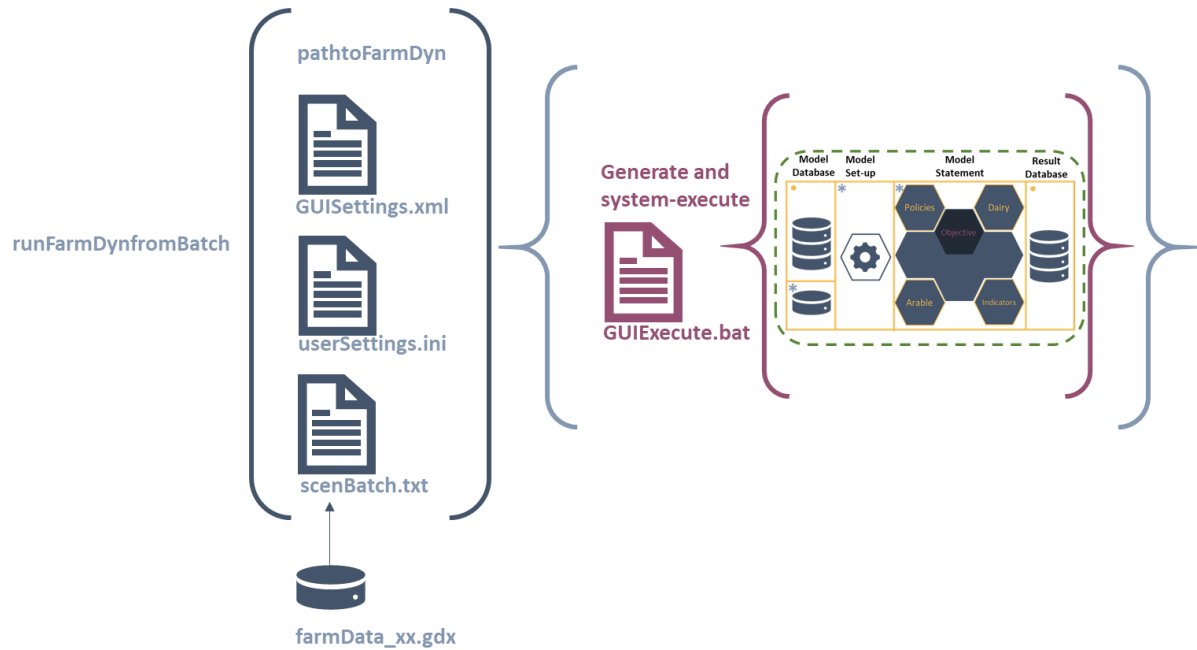


Figure 7 Schematic view of the wrapper “runFarmDynfromBatch”

4. CONCLUSIONS

The work-flow described in this deliverable 7.2 is an example for the integration of farm-level research activities within the MIND STEP project. It permits the exploitation of empirical results regarding risk attitudes of farmers by creating farm-specific parameterizations of a risk-utility function within the FarmDyn model. The handling of data and the execution of the FarmDyn model is facilitated by an R package FarmDynR, that contains a wrapper function for the execution of FarmDyn and provided further functionality for the aggregation of individual farm data. This general concept to prepare data and execute the model from an R environment, which is familiar to most model developers and empirical researchers in MIND STEP, will be extended also to the empirical work in Tasks 3.3 and 3.4.

5. ACKNOWLEDGEMENTS

This deliverable D7.2: “Beta version of a wrapper to allow a standardized communication channel between the models ” is developed as part of the H2020 MIND STEP project which received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 817566.

6. REFERENCES

Britz, W., 2014. A new graphical user interface generator for economic models and its comparison to existing approaches. *Ger. J. Agric. Econ.* 63, 271–285.

Britz, W., Kallrath, J., 2012. Economic simulation models in agricultural economics: the current and possible future role of algebraic modeling languages., in: Kallrath, J. (Ed.), *Algebraic Modeling Systems: Modeling and Solving Real World Optimization Problems*. Springer, Berlin, Heidelberg, pp. 199–212. <https://doi.org/10.1007/978-3-642-23592-4>

Britz, W., Lengers, B., Kuhn, T., Schäfer, D., 2016. A highly detailed template model for dynamic optimization of farms-FARMDYN. University of Bonn. Inst. Food Resour. Econ. Version Sept. 147.

Britz, W., Ciaian, P., Gocht, A., Kanellopoulos, A., Kremmydas, D., Müller, M., Petsakos, A., Reidsma, P., 2021. A design for a generic and modular bio-economic farm model. *Agric. Syst.* 191, 103133. <https://doi.org/10.1016/j.agsy.2021.103133>

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5(4), 297-323.

