GAMS code, modules & realizations

How do the magpie folder and code look like?

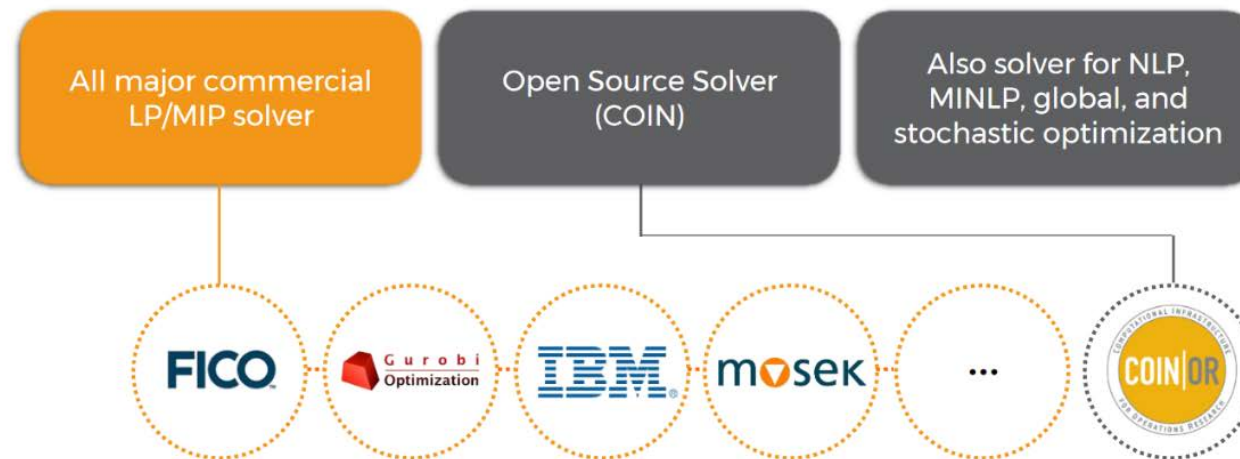**Presenter:** Isabelle Weindl

weindl@pik-potsdam.de

June 16, 2025

Outline

- What is GAMS?

- General structure of the MAgPIE Model

- **The magpie folder:** Components of MAgPIE, the structure of modules and realizations, and the full.gms file

- Coding etiquette: Variable and parameter naming

- Brief exercise

What is GAMS?

› MAgPIE is written in the **G**eneral **A**lgebraic **M**odeling **S**ystem **(GAMS)** language

*"GAMS is a high level modeling system for **mathematical programming** and **optimization**. It consists of a **language compiler** and a range of associated **solvers**."*


Useful links:

GAMS at a glance: **https://www.gams.com/products/gams/gams-language/**

GAMS documentation: https://www.gams.com/latest/docs/

General structure of the MAgPIE Model

- Future **population** and **GDP** are the main drivers (different **scenarios**).

- **Food consumption** patterns lead to **demand** for primary agricultural products.

- **Trade** patterns lead to regional **production.**

- **Production and biophysical** yield and water availability data shape **cropping patterns.**

- **Cropping patterns** drive **land use** decisions.

> These interactions and calculations are described in the modules of MAgPIE.

More info at: https://rse.pik-potsdam.de/doc/magpie/4.10.1/
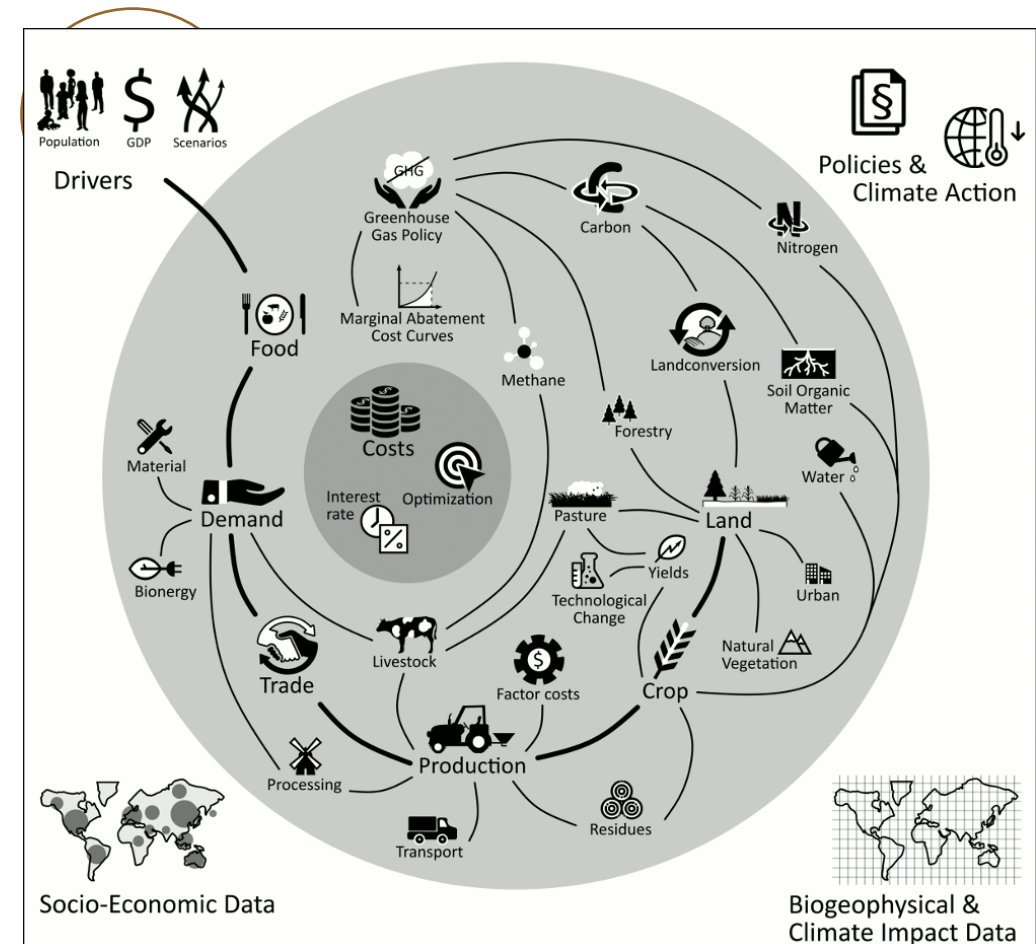
The magpie folder: Components of MAgPIE

After you have **cloned the magpie folder** (*git clone* [*https://github.com/magpiemodel/magpie.git*](https://github.com/magpiemodel/magpie.git)) from the repository, you will find :

The configuration settings are set in the default.cfg file (or in the start scripts).

46 modules and the include.gms file

Each folder within the modules folder represents a component of the model and contains different realizations (approaches to that component).

Structure of the modules

Each module (within the modules folder) **has a similar structure**:

Input folder with input files for all realizations	
Realization folders containing the source code of each realization	
The modules.gms file with the module description and listing of all realizations	

New realizations can be added by keeping the same structure. In that sense, MAgPIE is easily extendable.

Structure of realizations

As with the modules folders, the **realizations files (.gms) and folders are similarly constructed.** The source code is distributed over several gms-files to ensure the correct order of calculations during the optimization:

MAgPIE Workshop 2025 > WS25 model version > magpie > modules > 38_factor_costs > sticky_feb18

Name	Date modified	Type
input	15 Jun 2025 15:55	File folder
declarations.gms	15 Jun 2025 15:55	GMS File
equations.gms	15 Jun 2025 15:55	GMS File
input.gms	15 Jun 2025 15:55	GMS File
postsolve.gms	15 Jun 2025 15:55	GMS File
preloop.gms	15 Jun 2025 15:55	GMS File
presolve.gms	15 Jun 2025 15:55	GMS File
realization.gms	15 Jun 2025 15:55	GMS File
scaling.gms	15 Jun 2025 15:55	GMS File
sets.gms	15 Jun 2025 15:55	GMS File

gms.file	function
declarations.gms	Declares all variables, equations, and parameters for the realization.
equations.gms	Contains functional relationships that have to be fulfilled within the optimization.
input.gms	Loads input from `any_module/input` or `any_module/a_realization/input`.
sets.gms	Lists sets that are used (mainly) within this realization or are needed for interfaces defined within this realization.
preloop.gms	Includes calculations to be executed before the time-step loop starts.
presolve.gms	Includes calculations to be executed for each time step before the optimization process.
postsolve.gms	Includes calculations to be executed for each time step after the optimization process and defines output.
nl_fix.gms	Fixes non-linear behaviour to linear behavior.
nl_release.gms	Releases restrictions imposed during non-linear fixes.
scaling.gms	Lists the expected order of magnitude of specific variables calculated in this realization to improve the efficiency of the run.
not_used.txt	Lists interfaces (declared in other modules) that are not used within this realization, but in other realizations of the same module.

Note that not every gms-file is needed in every realization.

Coding etiquette: Variable and parameter naming

```
1   *** |  (C) 2008-2025 Potsdam Institute for Climate Impact Research (PIK)
2   *** |  authors, and contributors see CITATION.cff file. This file is part
3   *** |  of MAgPIE and licensed under AGPL-3.0-or-later. Under Section 7 of
4   *** |  AGPL-3.0, you are granted additional permissions described in the
5   *** |  MAgPIE License Exception, version 1.0 (see LICENSE file).
6   *** |  Contact: magpie@pik-potsdam.de
7
8   equations
9     q38_cost_prod_labor(i)          Regional labor input costs for crop production (mio USD17MER)
10    q38_cost_prod_capital(i)        Regional capital input costs for crop production (mio USD17MER)
11    q38_investment_immobile(j,kcr)  Cellular immobile investments into farm capital  (mio USD17MER)
12    q38_investment_mobile(j)        Cellular mobile investments into farm capital     (mio USD17MER)
13  ;
14
15  positive variables
16    vm_cost_prod_crop(i,factors)    Regional factor costs of capital and labor for crop production  (mio USD17MER  per yr)
17    v38_investment_immobile(j,kcr)  Investment costs in immobile farm capital (mio USD17MER per yr)
18    v38_investment_mobile(j)        Investment costs in mobile farm capital (mio USD17MER per yr)
19  ;
20
21  parameters
22    p38_labor_need(t,i,kcr)         Labor input costs per unit of output (USD17MER per ton DM)
23    p38_capital_need(t,i,kcr,mobil38)  Capital requirements per unit of output (USD17MER per ton DM)
24    p38_capital_immobile(t,j,kcr)   Preexisting immobile capital stocks before investment (mio USD17MER)
25    p38_capital_mobile(t,j)         Preexisting mobile capital stocks before investment (mio USD17MER)
26
27    p38_capital_cost_shares_iso(t,iso)   Capital shares of factor costs on iso level (1)
28    p38_capital_share_calibration(iso)   Summation factor used to calibrate calculated capital shares with historical values (1)
29    pm_factor_cost_shares(t,i,factors)   Capital and labor shares of factor costs on regional level (1)
30
31    p38_croparea_start(j,w,kcr)     Agricultural land initialization area (mio. ha)
32
33    i38_fac_req(t_all,i,kcr)        Factor requirements (USD17MER per tDM)
34  ;
35
36
37  *################## R SECTION START (OUTPUT DECLARATIONS) ##################
38  parameters
39    ov_cost_prod_crop(t,i,factors,type)    Regional factor costs of capital and labor for crop production  (mio USD17MER  per yr)
40    ov38_investment_immobile(t,j,kcr,type) Investment costs in immobile farm capital (mio USD17MER per yr)
41    ov38_investment_mobile(t,j,type)       Investment costs in mobile farm capital (mio USD17MER per yr)
```

q_ Equations

v_ Variables

p_ processing parameters

i_ input parameters

o_ output parameters

Prefixes are extended using either **m** (used in multiple modules/core code) or a **two digit number** (only used in the current module).

… In other files you will also find:
c_ Switches
s_ Scalars
f_ File parameters
m_ Macros

Coding etiquette: Variable and parameter naming

In other cases, the **prefixes are extended** with **a second letter** to indicate details such as:

?c_ current time step

`pcm_land(j,land)`

?q_ parameter containing the values of an equation

`oq10_land(t,j,type)`

?v_ parameter containing the values of a variable

`ov_landreduction(t,j,land,type)`

Suffixes indicate the **level of aggregation** of an object:

- (no suffix) Highest disaggregation available

- **_setname** aggregation **over set**

`i42_wat_req_k(t,j,k)`

- **_reg** **regional** aggregation

`v11_cost_reg(i)`

- **_glo** **global** aggregation

`i32_max_aff_area_glo`

The full.gms file

Once you have started a magpie run, you can find the full.gms file in the **ouput folder**.

Contains the final code used in the current MAgPIE run, based on the selected settings and on one realization per module (done automatically).

Brief exercise

(For online participants: Green tick when you are ready) ✅

1. Within the magpie folder, find the **14_yield** module.
2. Open the **declaration.gms** file of the **managementcalib_aug19** realization.
3. Find one example of:
   - A parameter used **only** within the current module
   - An equation
   - A processing parameter used **in** this module and the core code/another module
   - An output parameter

Thank you

weindl@pik-potsdam.de