GAMS CODE, MODULES & REALIZATIONS

Presenter: Edna J. Molina Bacca

mbacca@pik-potsdam.de

March 7th, 2022

Outline – GAMS code, modules, and realizations

- What is GAMS?
- General structure of the MAgPIE Model
- Structure of modules and realizations
- 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

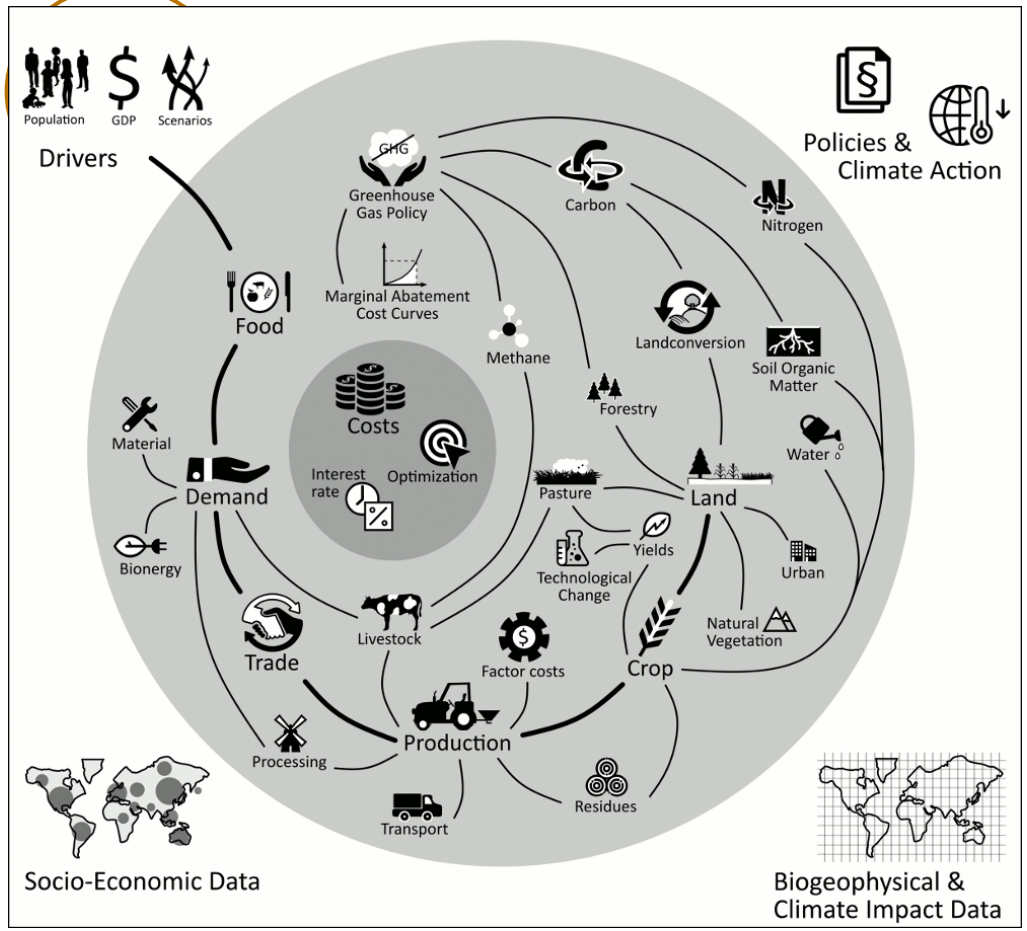
*"GAMS is a high level modeling system for **mathematical programming** and **optimization**. It consists of a **language compiler** and a range of associated **solvers**."*

Useful links:
GAMS at a glance: https://www.gams.com/products/gams/gams-language/
GAMS documentation: https://www.gams.com/35/docs/index.html

General structure of the MAgPIE Model

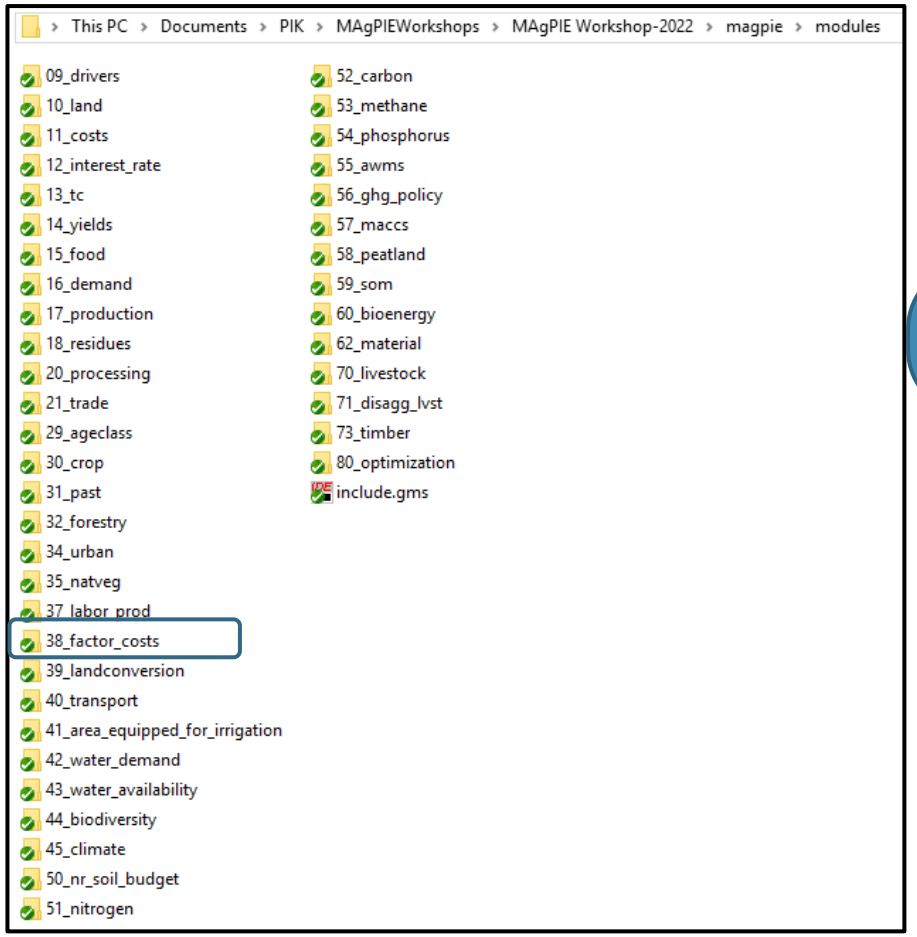
- Future **population** and **GDP** are the main drivers (different **scenarios**).
- **Food consumption** patterns lead to certain **demand**
- **Trade** patterns leads to regional **production**
- **Production and biophysical** yield and water availability data translate into **cropping patterns.**
- **Cropping patterns** drive **land use** decisions

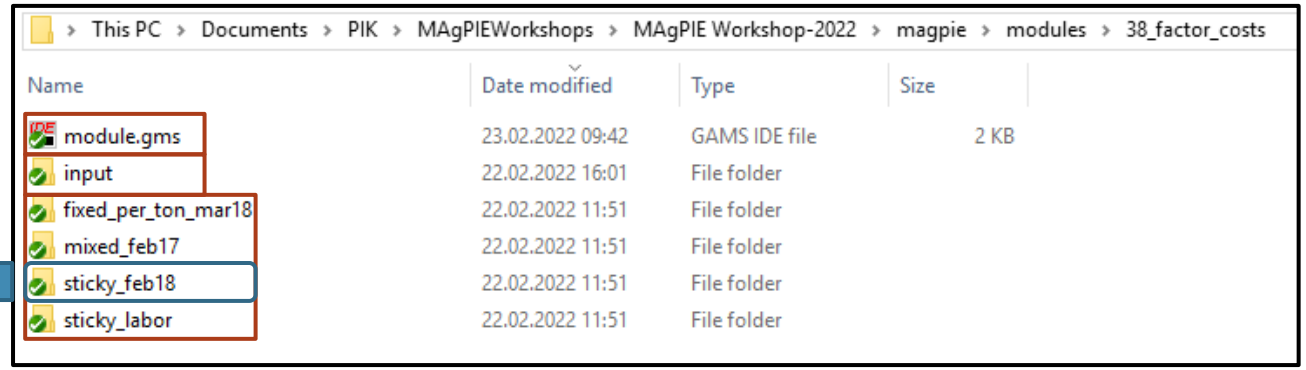
These interactions and calculations are described in MAgPIE's modules

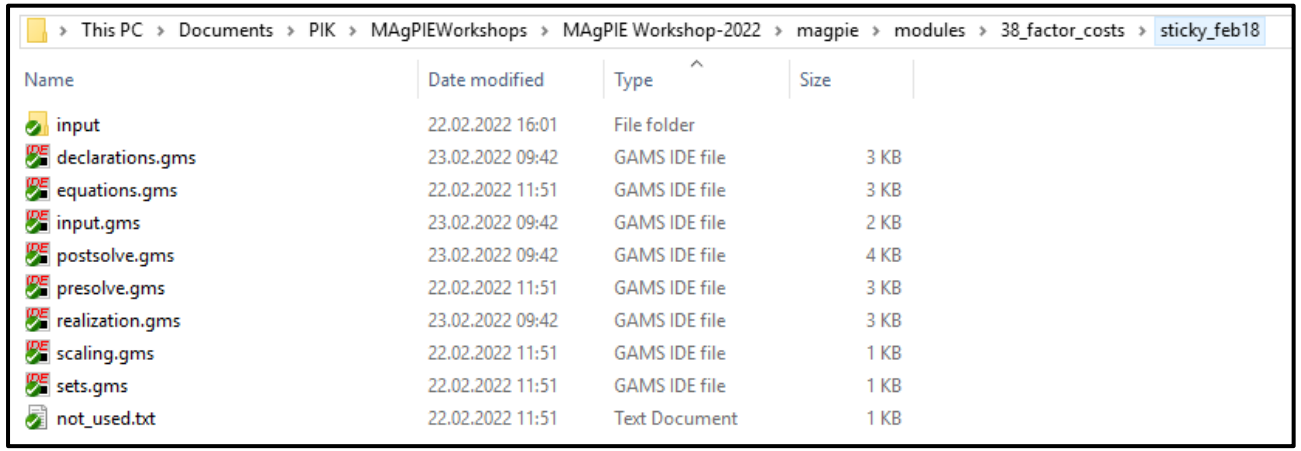


More info at: https://rse.pik-potsdam.de/doc/magpie/4.4.0/

Structure of modules and realizations

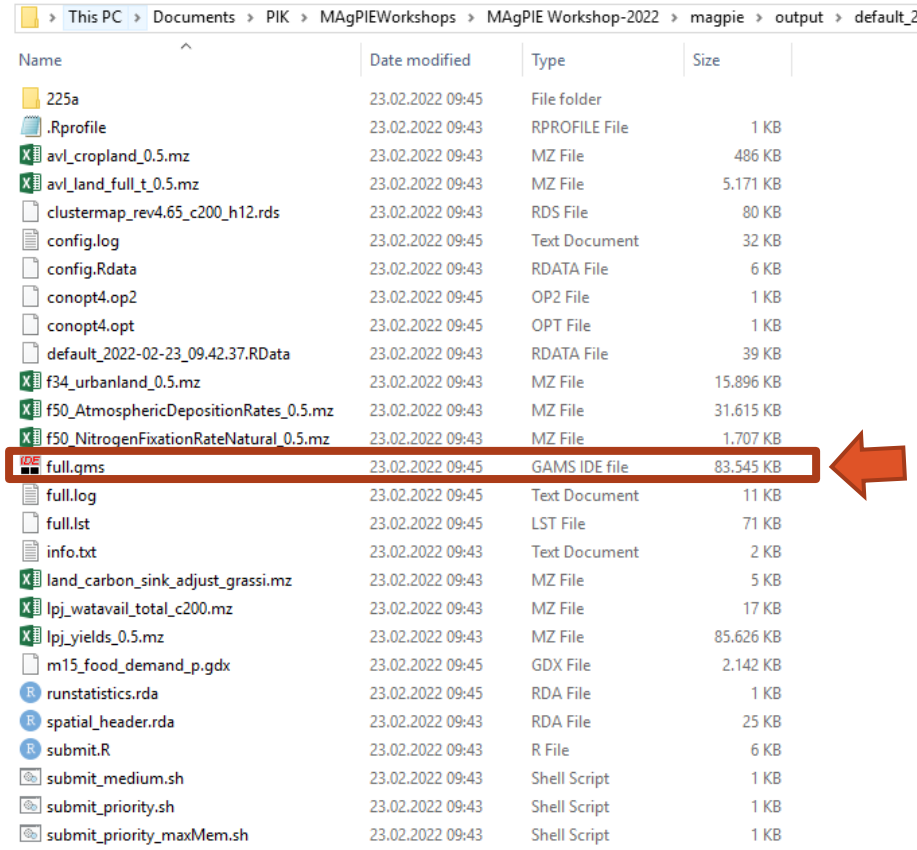






In other modules you can also find: preloop.gms, nl_fix.gms, and nl_release.gms.

Structure of modules and realizations



Contains the final code used in the current MAgPIE run, based on the selected settings and on one realization per module

Coding etiquette: Variable and parameter naming

```
equations
 q38_cost_prod_crop(i,kcr)          Regi
 q38_cost_prod_inv(i)               Regi
 q38_investment_immobile(j,kcr)     Cell
 q38_investment_mobile(j)           Cell
;

positive variables
 vm_cost_prod(i,kall)
 vm_cost_inv(i)
 v38_investment_immobile(j,kcr)
 v38_investment_mobile(j)
;

parameters
 p38_variable_costs(t,i,kcr)
 p38_capital_need(t,i,kcr,mobil38)
 p38_capital_immobile(t,j,kcr)
 p38_capital_mobile(t,j)

 p38_capital_cost_share(t,i)
 p38_share_calibration(i)

 p38_croparea_start(j,w,kcr)
;


*################### R SECTION START (
parameters
 ov_cost_prod(t,i,kall,type)
 ov_cost_inv(t,i,type)
 ov38_investment_immobile(t,j,kcr,type)
 ov38_investment_mobile(t,j,type)
 oq38_cost_prod_crop(t,i,kcr,type)
```

q_ equations

v_ variables

p_ processing parameters

o_ output parameter

```
$setglobal c38_sticky_mode  free
* options: dynamic, free

scalars
*' Depreciation rate assuming roughly 20 years linear depreciation for invesment goods
s38_depreciation_rate depreciation rate (share of costs)  / 0.05 /
*' Share of immobile capital.
s38_immobile  immobile capital (share) / 1 /
;

parameter f38_fac_req(kcr) Factor requirement costs in 2005 (USD05MER per tDM)
/
$ondelim
$include "./modules/38_factor_costs/sticky_feb18/input/f38_fac_req_fao.csv"
$offdelim
/
;
```

c_ switch

s_ scalars

f_ file parameter

… In other realizations you find:

`i30_avl_cropland_iso(iso)` i_ input parameters

`$macro m_timestep_length` m_ macros

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

Coding etiquette: Variable and parameter naming

Other extentions to prefixes:

- ?c_ current time step
- ?q_ parameter containing the value of an equation
- ?v_ paramter containing the values of a variable

```
oq10_land(t,j,type)
```

```
pcm_land(j,land)
```

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

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

- (no suffix) Highest disaggregation available
- **_setname** aggregation **over set**
- **_reg regional** aggregation
- **_glo global** aggregation

```
i42_wat_req_k(t,j,k)
```

```
v11_cost_reg(i)
```

```
i32_max_aff_area_glo
```

Exercise

(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. One example of:
   - Variable used **only** within the current module
   - An equation
   - A processing parameter used in **at least** this model and the core code.
   - An output parameter

THANK YOU

mbacca@pik-potsdam.de