

Introduction to GAMS Modeling Language

Part I

Dr. Alexander Gocht

Dr. Davit Stepanyan

Thuenen Institute of Farm Economics, Braunschweig

Structure of the Tutorial (part I)

- GAMS modeling language. General introduction; Introduction to GAMS IDE and Studio
- MyFarm LP model; Main elements of a GAMS model: variables, parameters, equations
- Coding MyFarm LP model in GAMS; Solver output
- Solver output; Improving efficiency. Sets, Subsets, Alias, Sum
- Introducing Sets and the Sum operator in MyFarm LP; Further useful Gams statements. Prod, Table, Variable-Attributes, Loop

Objectives

- Upon completion of this tutorial, the participants will be able to:
 - Use the GAMS Studio to modify or code economic simulation models in GAMS
 - Understand the logic behind the GAMS modeling language
 - Differentiate between the main elements of a GAMS model
 - Code simple economic models in GAMS
 - Debug these models
 - Analyze the result files generated by GAMS
 - Exchange data between GAMS and Microsoft Excel

References

- GAMS Documentation (2021). GAMS Development Corporation

What is GAMS?

- General Algebraic Modeling System (GAMS) is a high-level modeling system for mathematical programming and optimization.
- GAMS is “a tool for the development, solution, and management of large scale optimization problems”
- Their main distinguishing features are :
 - the use of relational algebra
 - and the ability to provide partial derivatives on multidimensional, very large and sparse structures
- GAMS enables the user to solve/optimize linear as well as non-linear equation systems
 - Optimization problems (maximization/minimization)
 - Fully determined equation systems
 - Combinations
- GAMS consists of a modeling language (along the lines of standard algebra) and solvers to solve or optimize equation systems.

Key Principles of GAMS

- The problem representation is independent of the solution method.
- The data representation follows the relational data model.
- The problem and data representations are independent of computing platforms.
- The problem and data representations are independent of user interfaces.
- Optimization methods will fail, and systems have to be designed to be fail-safe.

GAMS IDE vs. GAMS Studio

- GAMS IDE

- Written in Delphi
- In use for + 20 years
- Restricted to Windows



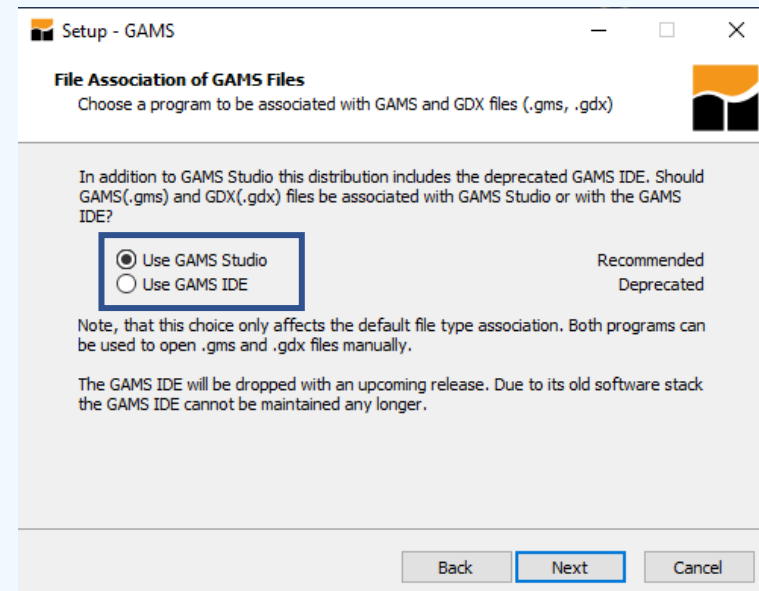
- GAMS Studio

- Written in C++
- Since 2019
- Setup similar to IDE
- Platform-independent (Windows, macOS, Linux)

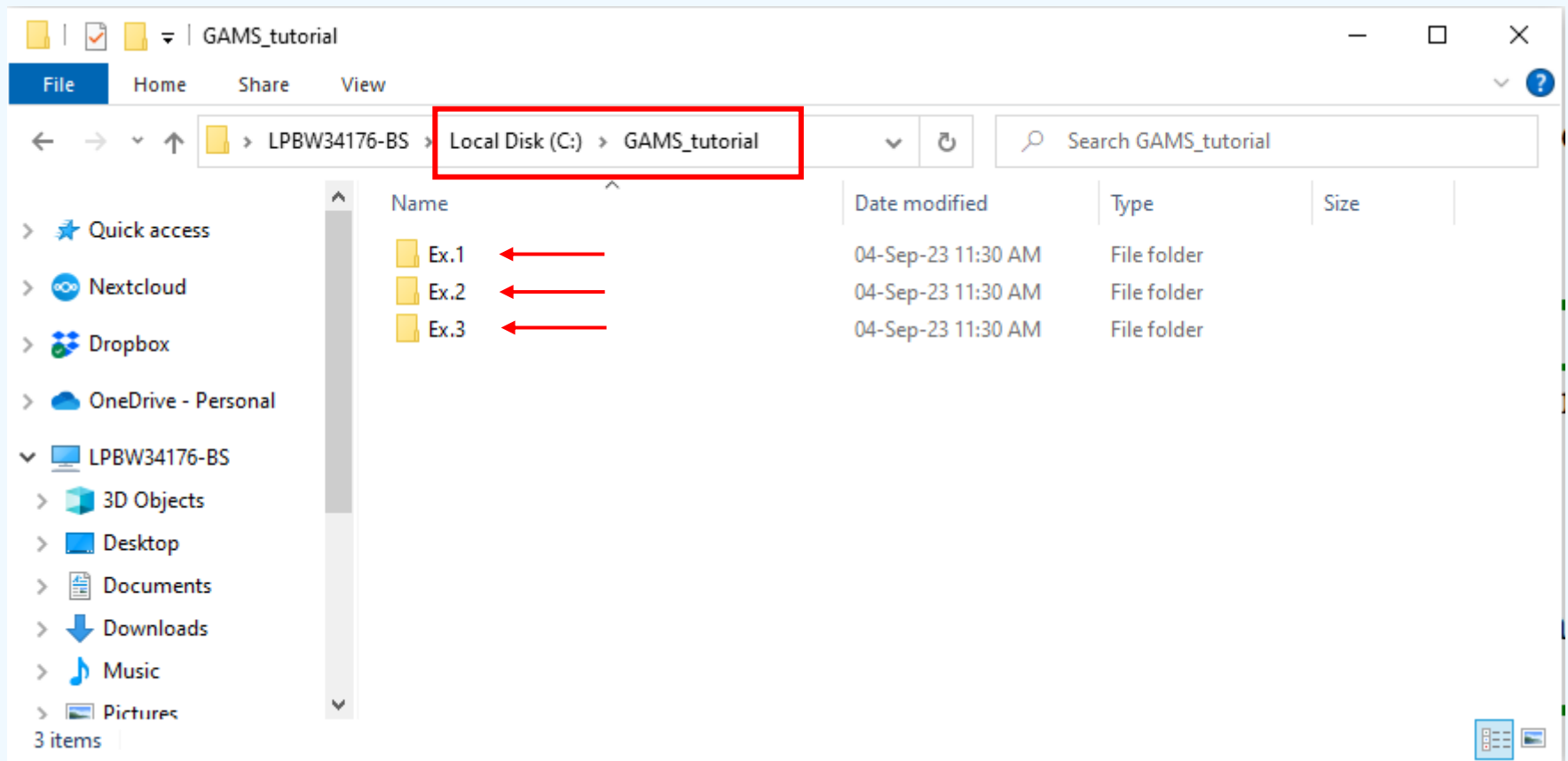


Installing GAMS

- Download the latest GAMS release:
<https://www.gams.com/download/>
- Install GAMS
 - Directory: `C:\GAMS\`
 - Choose GAMS Studio as default
- Create a directory for our class projects:
 - `C:\GAMS_Tutorial\`



Organization of the Files



GAMS Studio

Introduction



GAMS Studio: Welcome Page

The screenshot shows the GAMS Studio application window. The title bar reads "GAMS Studio" and the menu bar includes "File", "Edit", "GAMS", "MIRO", "Tools", "View", and "Help". The main area is divided into four panels: "Last Projects", "Last Files", "Getting Started", and "Further Help".

Last Projects:

- project1.gsp
- Exercise03.gsp
- Exercise02.gsp
- Ohne_Titel.gsp
- 4_LSS_BaseSim.gsp
- 2_LSS_Core.gsp
- 1_LSS_Input.gsp
- 1_LSS_input.gsp

Last Files:

- capmod_1.lst
- Exercise03.gms
- CarbonTax100_endotech_noc.gms
- Exercise02.gms
- metakiel_scenarios.gms
- header_capmod.gms
- ti_ecc10.gms
- capmod.gms
- Ohne_Titel.gms
- MyFarm_results.gdx
- Exercise05.gms

Getting Started:

- Create New File
- Browse GAMS Libraries
- Transport Example
- GAMS Studio Documentation
- GAMS Tutorial

Further Help:

- What's new in Studio?
- GAMS Release Notes
- GAMS World Forum
- Contact GAMS

At the bottom, there is a "Filename" field and a "Navigator: type '?' for help." field.

GAMS Studio: General Settings

- File -> Settings or [F7]

The image displays three screenshots of the GAMS Studio Settings dialog box, illustrating different configuration sections:

- General Settings:** Shows the default workspace path (C:/GAMS_Tutorial), options for showing the welcome page, restoring tabs, and saving files before running GAMS.
- Editor Settings:** Configures the editor's appearance and behavior, including font (Courier New), font size (10), indentation, line numbers, and completion settings.
- Default Settings:** Controls the symbol view (List View), attributes (Level, Marginal, Lower Bound, Upper Bound, Scale), and preferences like trailing zeroes and decimal separators.

Exercise 1. Linear programming model

LP Model MyFarm



[This Photo](#) by Unknown Author is licensed under [CC BY](#)

A linear Programming (LP) Problem: “Myfarm” Example

	Wheat	Barley	Rapeseed	Sugarbeet
Gross margin (€/ha)	253	443	284	516
Labor requirment (hours/ha)	25	36	27	87

- Farms size: 200 ha
- Labor availability: 10000 hours
- X_i : area of land devoted to each crop
- $X_i \geq 0$
- Maximize profit

The Mathematical Model

$$\mathbf{Max! Z} = 253 * X_{\text{wheat}} + 443 * X_{\text{barley}} + 284 * X_{\text{rapeseed}} + 516 * X_{\text{sugarbeet}}$$

• Subject to:

• $X_{\text{wheat}} ; X_{\text{barley}} ; X_{\text{rapeseed}} ; X_{\text{sugarbeet}} \geq 0$ (non-negativity)

• $X_{\text{wheat}} + X_{\text{barley}} + X_{\text{rapeseed}} + X_{\text{sugarbeet}} \leq 200$ (land)

• $25 * X_{\text{wheat}} + 36 * X_{\text{barley}} + 27 * X_{\text{rapeseed}} + 87 * X_{\text{sugarbeet}} \leq 10000$ (labor)

• Where:

✓ X_{wheat} : land area devoted to wheat production

✓ X_{barley} : land area devoted to barley production

✓ X_{rapeseed} : land area devoted to rapeseed production

✓ $X_{\text{sugarbeet}}$: land area devoted to sugar beet production

MyFarm in Excel

Hands-on exercise

- Open the **Exercise 01-MyFarm in Excel.xlsx** file and solve the LP problem

Questions

- The area devoted to wheat is ___ha. Why?
- Are the land and labour resources fully exploited?
- The max. total gross margin is _____€

The Structure of GAMS Models

Main Elements of a GAMS Model

- 3 essential parts to formulate a GAMS model:
 - Variables
 - Parameters
 - Equations
- To use these in GAMS, 2 steps are required:
 1. Declaration: give it a name and tell GAMS what it is, i.e., parameter, variable, equation, ...
 2. Assignment or definition (a specific value, type, or function)
- Of course, many more statements are available

GAMS Statement: Variables

- Entities whose values are generally unknown until after a model has been solved
- A GAMS variable, like all other identifiers, must be declared before it may be referenced.
- The syntax

```
Variables v_obje;
```

```
Positive variables v_actLevlWHEAT, v_actLevlBARLEY, v_actLevlRAPESEED, v_actLevlSUGARBEET;
```

- **Variable(s)** Keyword for variable definition
- **Positive/Binary** Keyword can be preceded by modifier:
 - **Positive** The variable can only contain nonnegative values
 - **Binary** Only 0 and 1 allowed
- v_actLevlWHEAT, ... List of variable identifiers
- ; Semicolon ends each GAMS statement

GAMS Statement: Variables. Syntax

```
[var_type] variable[s] var_name [text]
```

Keyword	Description	Default Lower Bound	Default Upper Bound
free (default)	No bounds on variable. Both bounds may be changed from the default values by the user.	-inf	+inf
positive or nonnegative	No negative values are allowed for variable. The user may change both bounds from the default value.	0	+inf
negative	No positive values are allowed for variables. The user may change both bounds from the default value.	-inf	0
binary	Discrete variable that can only take values of 0 or 1. For details see section Types of Discrete Variables . In relaxed Model types the integrality requirement is relaxed.	0	1
integer	Discrete variable that can only take integer values between the bounds. The user may change both bounds from the default value. The default upper bound inside GAMS is +inf but when the variable is passed on to the solver, the option or command line parameter IntVarUp decides what upper bound (by default 100) is passed on to the solver in case GAMS has upper bound +inf. In relaxed Model types the integrality requirement is relaxed.	0	+inf
sos1	A set of variables, such that at most one variable within a group may have a non-zero value. For details see section Types of Discrete Variables .	0	+inf
sos2	A set of variables, such that at most two variables within a group may have non-zero values and the two non-zero values are adjacent. For details see section Types of Discrete Variables .	0	+inf
semicont	Semi-continuous, must be zero or above a given minimum level. For details see section Types of Discrete Variables .	1	+inf
semiint	Semi-integer, must be zero or above a given minimum level and integer. For details see section Types of Discrete Variables . The default upper bound inside GAMS is +inf but when the variable is passed on to the solver, the option or command line parameter IntVarUp decides what upper bound (by default 100) is passed on to the solver in case GAMS has upper bound +inf. In relaxed Model types the integrality requirement is relaxed.	1	+inf

GAMS Statement: Variables

- Good modeling practice:
 - Use expressions which are short (one word) but are still telling
 - Not VAR1, A, B
 - Add explanatory text to further clarify the meaning:

```
Variables
v_obje          objective function value
;
Positive variables
v_actLevlWHEAT  land area wheat
v_actLevlBARLEY land area barley
v_actLevlRAPESEED land area rapeseed
v_actLevlSUGARBEET land area sugar beet
;
```

GAMS Statement: Parameters

- Are constants and contain exogenously given values
- Are not modified during the solution process

Parameters

```
p_uvag_wheat      Gross margin of wheat      /253/,  
p_uvag_barley     Gross margin of corn      /443/,  
p_uvag_rapeseed   Gross margin of rapeseed   /284/,  
p_uvag_sugarbeet  Gross margin of sugar beet   /516/  
;
```

• Where

- `Parameters` Keyword for parameter definition
- `p_uvag_wheat` Identifier of the parameter
- `/253/` Initial assignment of a value (otherwise: 0)
- `,` To define more parameters in one statement, separate them by commas or breaks
- `;` Every GAMS statement is concluded by a semicolon

GAMS Statement: Parameters

- Good modeling practice: Separate declaration and assignment

```
Parameters
p_uvag_wheat      Gross margin of wheat
p_uvag_barley     Gross margin of barley
;
p_uvag_wheat = 253;
p_uvag_barley = 443;
```

- Assignment may contain arithmetic operations, e.g.

Operation	Symbol	Order of Precedence
Exponentiation	**	1
Multiplication	*	2
Division	/	2
Addition	+	3
Subtraction	-	3

GAMS Statement: Parameters

- Values of parameters can be “overwritten”:

```
Parameters
p_uvag_wheat      Gross margin of wheat
gm_barley         Gross margin of barley
;
p_uvag_wheat  = 253;
p_uvag_barley = 443;
p_uvag_wheat = 378;
```

GAMS Statement: Parameters

- To check the value of a parameter or variable use the display statement:

```
DISPLAY p_uvag_wheat;
```

or

```
Display p_uvag_wheat;
```

or

```
display p_uvag_wheat;
```

or

```
displAy p_uvag_wheat;
```

GAMS Statement: Equations

- In GAMS, each equation consists of 2 separate statements:
 - Declaration (declare the equations existence)
 - Definition (the equation itself, its algebraic form)

- Declaration:

```
Equations e_land, e_labour, obje;
```

- Where

- `Equations` Keyword for equation declaration
- `e_land, ... obje` List of equations to be declared
- `;` End GAMS statement

GAMS Statement: Equations

- Definition

- Equation name followed by two dots (..)
- Then the algebraic form of the equation
- In equations, the relational operators ($=$ \leq \geq) must be written as:

=E= Equality: right-hand side must equal left-hand side

=G= Greater than: left-hand side must be greater than or equal to right-hand side

=L= Less than: left-hand side must be less than or equal to right-hand side

GAMS Statement: Equations

- Definition
 - Equation definitions for the MyFarm-exercise:

```
obje ..      v_obje =E= p_uvag_wheat * v_actLevlWHEAT + p_uvag_barley *  
v_actLevlBARLEY + p_uvag_rapeseed * v_actLevlRAPESEED + p_uvag_sugarbeet *  
v_actLevlSUGARBEET;
```

```
e_land ..    v_actLevlWHEAT + v_actLevlBARLEY + v_actLevlRAPESEED +  
v_actLevlSUGARBEET =L= 200;
```

```
e_labour ..  p_lab_wheat * v_actLevlWHEAT + p_lab_barley * v_actLevlBARLEY +  
p_lab_rapeseed * v_actLevlRAPESEED + p_lab_sugarbeet * v_actLevlSUGARBEET =L= 10000;
```

GAMS Statement: Equations

- **Definition**

- **Note:** The general form of these statements is

```
equationname.. algebra1 relation algebra2;
```

- **Where**

- `equationname` The identifier of the equation as declared
- `..` Separator between name and equation
- `algebra1, algebra2` Some algebraic expressions containing parameters and at least one endogenous variable
- `relation` One of the following =E= or =L= or =G=
- `;` End of the statement

GAMS Statement: Model

- Once all the model structural elements have been defined, the model has to be defined by a `Model` statement to identify the equations that belong to the model.

```
Model myfarm /e_land, e_labour, obje/;
```

Or

```
Model myfarm /all/;
```

- **Where**
 - `Model` Keyword for model definition
 - `Myfarm` Identifier for this model
 - `/e_land, .../` List of equations that belong to this model
 - `;` Ends the statement
- The keyword `all` includes all previously defined equations in the model.

GAMS Statement: Solve

- The `Solve` statement causes GAMS to use a solver to optimize/solve the model

```
Solve myfarm using lp maximizing v_obje;
```

- **Where**
 - `solve` Keyword for the solve statement
 - `myfarm` Name of the model to be solved
 - `using lp` Declares type of solver to be used (lp = linear programming)
 - `maximizing` Declares the direction of the optimization (alternative: minimizing)
 - `v_obje` Target variable (has to be defined before and has to occur in the model, must not be restricted)
 - `;` Ends the statement

Good Modelling Practices

- Enhance the readability of the model (to others/ to you after some time of absence) by:
 - Ordering: e.g. first define all parameters, variables,...then assign them
 - Giving telling names to model-entries (“p_uvag_wheat” instead of “par_1”)
 - Defining all entries using explanatory text
 - Specify units of all entries (e.g. ha, 1000 USD,...)
 - Use comments wherever useful:

```
* starts a comment-line  
  
$ontext  
starts a comment which can stretch over several lines and needs to  
be ended by  
$offtext
```

Exercise 2. Linear programming model in GAMS

Coding the MyFarm Model



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

MyFarm Model

- The problem

Farmer wants to maximize his profit given 100 ha of land and 500 hours of labor. He/she has the option of cultivating wheat, barley rapeseed and/or sugarbeet. How much of each crop should be plated in order to maximize the profit.

	Wheat	Barley	Rapeseed	Sugarbeet
Gross margin (€/ha)	253	443	284	516
Labor requirement (hours/ha)	25	36	27	87

Questions

- The value of an additional unit of land is _____€
- If we include 1 ha of wheat in the cropping mix, the tot. gross margin will increase/decrease by _____€

Solver Output: the Listing File

The Listing File “.lst”

- The output file generated from a GAMS run is called listing file
- The listing file contains (in standard settings)
 - Echo print/ Compilation
 - Error messages
 - Equation listing
 - Column (variable) listing
 - Model statistics
 - Solution report
 - SolEQU, SolVAR:
 - Solution values for equations and variables

The Listing File “Echo print”

- Is always the first part of the output file
- It is a listing of the input with added line numbers

```
22 *$title MyFarm
23 Variables
24   v_obje  objective function value
25 ;
26
27 Positive variables
28   v_actLevlWHEAT    land area wheat
29   v_actLevlBARLEY  land area barley
30   v_actLevlRAPESEED land area rapeseed
31   v_actLevlSUGARBEET land area sugarbeet
32 ;
33
34 Parameters
35   p_uvag_wheat      Gross margin of wheat      /253/ ,
36   p_uvag_barley     Gross margin of barley     /443/ ,
37   p_uvag_rapeseed   Gross margin of rapeseed   /284/ ,
38   p_uvag_sugarbeet  Gross margin of sugarbeet  /516/ ,
39 ;
40   p_lab_wheat       Required labor hours of wheat /25/ ,
41   p_lab_barley      Required labor hours of barley /36/ ,
42   p_lab_rapeseed    Required labor hours of rapeseed /27/ ,
43   p_lab_sugarbeet   Required labor hours of sugarbeet /87/ ,
44 ;
45
46
47 Equations
48 *Declaration
49
50   e_land  land constraint
51   e_labour labour constraint
52   obje    objective function
53 ;
54
55 *Defintion
56 obje ..      v_obje =E= p_uvag_wheat * v_actLevlWHEAT + p_uvag_barley * v
_actLevlBARLEY + p_uvag_rapeseed * v_actLevlRAPESEED + p_uvag_sugarbeet *
v_actLevlSUGARBEET;
57
58 e_land ..    v_actLevlWHEAT + v_actLevlBARLEY + v_actLevlRAPESEED + v_act
LevlSUGARBEET =L= 200;
59
60 e_labour ..  p_lab_wheat * v_actLevlWHEAT + p_lab_barley * v_actLevlBARLE
Y + p_lab_rapeseed * v_actLevlRAPESEED + p_lab_sugarbeet * v_actLevlSUGARB
EET =L= 10000;
61
62
63
64 Model myfarm /all/;
65
66 Solve myfarm using lp maximizing v_obje;
```

The Listing File “Error messages”

- All errors are marked by 4 stars (****)
 - Search for **** to find errors in the listing file
- Two types of error messages:
 1. **Compilation errors** (syntax/consistency mistakes)
 - `$errornumber` is placed below the exact position in the line where the error occurred
 - `errornumber` is referenced by an error listing that describes this error
 2. **Execution errors** (illegal arithmetic operations)
 - Errors after compilation finished, i.e., during model generation and solving

The Listing File “Error messages” Examples

- **Compilation error**
 - A dollar symbol and error number are printed below the offending symbol on a separate line that begins with four asterisks

```
55 *Defintion
56 obje ..      v_obje =E= p_uvag_whea * v_actLevlWHEAT + p_uvag_barley * v_
****
                        $140
                        actLevlBARLEY + p_uvag_rapeseed * v_actLevlRAPESEED + p_uvag_sugarbeet * v
                        _actLevlSUGARBEET;
**** 140 Unknown symbol
```

- **Execution error**
 - Occurs after compilation has finished

```
58 e_land ..    v_actLevlWHEAT + v_actLevlBARLEY + v_actLevlRAPESEED + v_act
LevlSUGARBEET =L= 200/0;
```

```
**** Exec Error at line 58: division by zero (0)
```

The Listing File “Equation listing”

- Is the first part of the output generated by a solve statement
- By default, the first three equations in every block are listed
 - This can be modified with the option `limrow`

```
Equation Listing      SOLVE myfarm Using LP From line 66

---- e_land  =L=  land constraint
e_land..  v_actLevlWHEAT + v_actLevlBARLEY + v_actLevlRAPESEED
          + v_actLevlSUGARBEET =L= 200 ; (LHS = 0)

---- e_labour =L=  labour constraint
e_labour.. 25*v_actLevlWHEAT + 36*v_actLevlBARLEY + 27*v_actLevlRAPESEED
           + 87*v_actLevlSUGARBEET =L= 10000 ; (LHS = 0)

---- obje  =E=  objective function
obje..  v_obje - 253*v_actLevlWHEAT - 443*v_actLevlBARLEY
        - 284*v_actLevlRAPESEED - 516*v_actLevlSUGARBEET =E= 0 ; (LHS = 0)
```

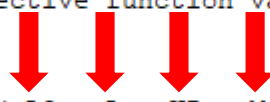
- For equalities: LHS should be equal to RHS, typically 0.

The Listing File “Column listing”

- The column listing or variable listing is the next part of the output

```
Column Listing      SOLVE myfarm Using LP From line 66

|---- v_obje  objective function value
v_obje
      1      obje      (.LO, .L, .UP, .M = -INF, 0, +INF, 0)
```



The Listing File “Model statistics”

```
Model Statistics      SOLVE myfarm Using LP From line 59

MODEL STATISTICS

BLOCKS OF EQUATIONS      3      SINGLE EQUATIONS      3
BLOCKS OF VARIABLES      5      SINGLE VARIABLES      5
NON ZERO ELEMENTS      13

GENERATION TIME      =      0.015 SECONDS      3 MB 39.2.1 98a2c774 WEX-WEI
```

The Listing File “Solution report”

- It is marked with the title Solution Report and includes the **solve summary**, the **solver report**, the **solution listing**, and the **report summary**

```
General Algebraic Modeling System
Solution Report      SOLVE myfarm Using LP From line 59

                S O L V E      S U M M A R Y

→ MODEL      myfarm           → OBJECTIVE      Z
→ TYPE       LP                → DIRECTION    MAXIMIZE
→ SOLVER     CPLEX             → FROM LINE    59

**** SOLVER STATUS      1 Normal Completion
**** MODEL STATUS      1 Optimal
**** OBJECTIVE VALUE          92607.8431

RESOURCE USAGE, LIMIT          0.016 10000000000.000
ITERATION COUNT, LIMIT        2      2147483647
```

The Listing File “The Solution Listing”

- The solution listing is a row-by-row then column-by-column listing of the solutions returned to GAMS by the solver program
- Each individual equation (SolEQU) and variable (SolVAR) is listed with four pieces of information
 - May be suppressed by

```
option solprint = off ;
```

- SolEQU:

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU e_land	-INF	200.000	200.000	391.471
---- EQU e_labour	-INF	10000.000	10000.000	1.431
---- EQU obje	.	.	.	1.000

Shadow price

The Listing File “The Solution Listing”

- SolVAR:

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR v_obje	-INF	92607.843	+INF	.
---- VAR v_actLevl~	.	.	+INF	-174.255
---- VAR v_actLevl~	.	145.098	+INF	.
---- VAR v_actLevl~	.	.	+INF	-146.118
---- VAR v_actLevl~	.	54.902	+INF	.

Reduced cost

The Listing File “Report Summary”

```
**** REPORT SUMMARY :           0      NONOPT  
                                0 INFEASIBLE  
                                0  UNBOUNDED
```