GAMS-Programming

November 9, 2015

A casual gams extends of various section with a spacific order. In the following these sections are summed up with both, syntax and semantics. The GAMS-program has one of the following styles.

1. Data:
   (a) Set Declarations
   (b) Parameter declarations and Definitions
   (c) Assignments
   (d) Displays
2. Model
   (a) Variable Declaration
   (b) Equation Declaration
   (c) Equation Definition
   (d) Model Definition
3. Solution
   (a) Solve
   (b) Displays

The second style of a GAMS-program is under construction

1 Data

1.1 Set Declarations

In the *set*-declaration the different sets that are needed to solve the LP are declared. This means here you name all the different sets that are requiered to solve the LP.

```
Sets
     i    1 / 2 / 3 /
     j    berlin / aachen / hamburg / ;
```

In `GAMS` there are some rules for `Sets`.

1. after each element of a set a / must follow
2. end the `Sets`-section with a ";"

1.2 Parameter declarations and Definitions

In the *parameter*-declaration functions of the LP are declared and defined. These functions mostly depend on at least one set. For the input (set elements) the *parameter* are fixed values.

```
Parameters
     f(x)      some  function
       /       x1      1
               x2      2 / ;
```

The rules for `Parameters` are the following:

1. split different paramteters with /
2. end the `Parameters`-section with ";"

1.3 Assignments

Under Construction

1.4 Displays

Under Construction

2 Model

2.1 Variable Declaration

In this part of the program *variables* can be declared. The *variables* represent the *decision variables* of the LP.

```
Variables
     x(i)  some function depending on i
```

2.2 Equation Declaration

The *equations* of a GAMS-program represent the constraints of the LP. But before defining the different constraints they are named. In case all equations are named a ";" marks the point where the equations are defined.

```
Equations
    equationName      A short description about the equation;
```

2.3 Equation Definition

After a *Equation* is declared it can be defined. To do so the following three expressions can be used: The definition of the equation are directly below the declaration of the *equation*. The ";" is used to

$\leq$	=l=
$=$	=e=
$\geq$	=g=

seperated the different equation. In order to define a equation the following syntax is used:

```
equationName  ..    z =g= 0;
```

2.4 Model Definition

3 Solution

3.1 Solve

3.2 Displays