

# Selected Advanced GAMS Features

## 1. The ALIAS Statement

- interaction between elements of the same set (e. g. specification of intermediate inputs)

*Alias (f,ff) ;*

## 2. The Dollar Operator

- definition of exceptions in indexed assignment statements or in indexed equation definitions

(1) The result of a logical operation is zero if the assertion is false and one if it is true.

(2) GAMS lacks an explicit Boolean data type. When interpreting the result of a logical condition GAMS takes zero as false and non-zero as true.

*C0(I)\$ (B0(I) NE 0) = A0(I)/B0(I) ;*

*C0(I)\$B0(I) = A0(I)/B0(I) ;*

- implicit if-else statements

*C0(I) = A0(I)\$ (Flag)+ B0(I))\$ (NOT Flag) ;*

- definition of exceptions in indexed equation definitions

*EQU(I)\$ (SIGMA(I) NE 1) .. Q(I)=E= (1/A(I))\* ( alpha(I)\*r\*\*(1-sigma(I))+  
(1-alpha(I))\*w\*\*(1-sigma(I)) )\*\*(1/(1-sigma(I))) ;*

### 3. Error Checks: ABORT and \$EXIT Statements

- Abort: Conditional abortion of compilation (e.g. data consistency checks)

```
SETS
    R      rows of SAM      /R1*R5/,
    C      columns of SAM   /C1*C5/;

ALIAS (R,RR), (C,CC);

TABLE SAM(R,C)   Social Accounting Matrix
      C1      C2      C3      C4      C5
R1      1      5      3      4      2
R2      2      4      1      5      3
R3      3      1      4      2      5
R4      4      2      5      3      1
R5      5      3      2      1      4 ;

*      Consistency check of Social accounting matrix

PARAMETER CHECK          Data check parameter;

PARAMETER CHECK(R,C);
CHECK(R,C)$(ORD(R) EQ ORD(C)) = SUM(CC, SAM(R,CC)) - SUM(RR, SAM(RR, C));

DISPLAY CHECK;

ABORT$(SMAX( (R,C), ABS(CHECK(R,C))) GE 1.E-6)"SAM DATA IS NOT CONSISTENT:",CHECK;
```

- \$EXIT: statement effects an exit from compilation. This statement is often used as a "quick and dirty" way to trace back compilation errors.

```
DISPLAY "This is a check for data consistency:", CHECK;
$EXIT
```

### 4. Inclusions of External Files: \$INCLUDE

- Insert the contents of an exogenous file within the main GAMS program.
- Separation of data and program
- Compilation errors within \$INCLUDE files will be pinpointed with the local line number in the respective file.

## 5. The LOOP Statement

- Execution of the statements within the scope of the loop for each member of the controlling set(s).

Example: Sequence of tax rates

```
PARAMETERS TAX(I) Ad-valorem tax;
```

```
EQU6(I).. X(I) =E= [ BETA(I)*M/(PC(I)*(1+ TAX(I))) ];
```

the iterative GAMS procedure could then be specified as follows:

```
SETS SC Scenario /T05, T10, T50, T100/
```

```
PARAMETERS TAXRATE(SC) Counterfactual ad-valorem taxes
/T05 0.05,
T10 0.1,
T50 0.5,
T100 1/;
```

```
LOOP(SC,
    TAX("1") = TAXRATE(SC);
SOLVE MODEL_2X2 USING NLP MAXIMIZING DUMMY;
);
```

## 6. The PUT Statement

- Output of results to different files at one time
- First specify the file where the GAMS output should be directed to using the FILE statement
- PUT statement (opposite to DISPLAY statement) only works on scalars and not on blocks → Use of LOOP whenever we want to write blocks

```
OUTPUT("SUPPLY",I) = Y.L(I);  
OUTPUT("DEMAND",I) = X.L(I);  
OUTPUT("PRICES",I) = PC.L(I);
```

```
DISPLAY OUTPUT;
```

```
*      Specify the name of the external results file  
FILE RES /RESULTS.DAT/;
```

```
*      Specify a set with the names of the items to be plotted as part of the report
```

```
SET    ITEMS /SUPPLY, DEMAND, PRICES/
```

```
*      Indicate to which file we plot now (there might be various output files in  
*      use)
```

```
PUT RES;
```

```
LOOP( (ITEM,I),  
      PUT ITEM.TL, OUTPUT(ITEM,I);  
      PUT /;  
    );
```

## 7. Environment Variables: \$SETGLOBAL

- Set an environment variables with the \$SETGLOBAL statement followed by a variable name and a specified value
- The variable can be referenced within the GAMS program by the expression *%variable name%*.

Important application: Environment variables are very useful to run a GAMS model for alternative settings of key parameters from batch (sensitivity analysis).

```
goto start
echo =====
echo Syntax:          run_sc kbar lbar
echo =====
echo.
echo where "kbar" is:
echo.
echo positive integer number (capital endowment)
echo.
echo and "lbar" is one of:
echo.
echo positive integer number (labor endowment)
echo.

:start
@echo off
echo $setglobal KBAR %1          >sc.inp
echo $setglobal LBAR %2          >>sc.inp

call gams mod_2x2.gms
```

By typing *run\_sc 30 50* on the DOS prompt we effect that the following two lines are written to the file *sc.inp* is written.

```
$setglobal KBAR 30
$setglobal LBAR 50
```

The file *sc.inp* can be included within the core model code by means of the \$INCLUDE statement. There we can use *KBAR* and *LBAR* as global variables and assign their values to the respective SCALARS.

```

$title: 2x2x1_MODEL

*      Set declaration and assignment
SETS
      I                goods /1 good_1, 2 good_2/;

$INCLUDE sc.inp

*      Data declaration and assignment
PARAMETERS
      A(I)              efficiency parameter /1 1.96, 2 1.889 /,
      ALPHA(I)          capital value share  /1 0.625, 2 0.667/,

      BETA(I)           expenditure share;

SCALARS
      KBAR              capital endowment    /%KBAR%/ ,
      LBAR              labor endowment     /%LBAR%/ ;

```