CHAPTER 5. A SIMPLE GENERAL EQUILIBRIUM MODEL IN GAMS

This chapter is based on two paper by Hans Lofgren: "Exercises in general equilibrium modelling using GAMS"and "Key to exercises in CGE modelling using GAMS". The first paper is a series of exercises bearing on the building and the use of a CGE model, the second paper includes the solutions of the exercises. You can download the papers at
http://www.ifpri.org/sites/default/files/publications/gamsexer.pdf
http://www.rri.wwu.edu/CGECourse/GAMSkey.pdf

The tutorial Chapter 2 in the user's guide is a short and clear presentation of GAMS code. This chapter presents a linear programming problem, and not a CGE model simulation problem. But it does nor matter to learn GAMS syntax.

1. A verbal model presentation

The model assumes that producers maximise profits subject to production functions, with primary factors as arguments, while households maximise utility subject to budget constraints. Cobb-Douglas functions are used both for producer technology and the utility functions from which household consumption demands are derived. Factors are mobile across activities, available in fixed supplies, and demanded by producers at market-clearing prices (rents). On the basis of fixed shares (derived from base-year data), factor incomes are passed on in their entirety to the households, providing them with their only income. The outputs are demanded by the households at market-clearing prices.

The model satisfies Walras' law in that the set of commodity market equilibrium conditions is functionally dependent. Any one of these conditions can be dropped. The proposed model drops the equilibrium condition for the nonagricultural commodity. The model is homogeneous of degree zero in prices. To assure that only one solution exists, a price normalisation equation, in this case fixing the consumer price index (CPI), has been added. After these adjustments, the model has an equal number of endogenous variables and independent equations. Given this definition of the price normalisation equation, all simulated price changes can be directly interpreted as changes vis-à-vis the CPI. The model is disaggregated into two households (urban and rural), two factors (labour and capital), and two activities and associated commodities (agriculture and nonagriculture). The model does not include a government, intermediate demands, savings, investment, or an outside world.

2. Mathematical model statement

All endogenous variables are written in uppercase Latin letters, whereas parameters (including variables with fixed or exogenous values) have lower-case Latin or Greek letters. Subscripts refer to set indexes, with one letter per index. Superscripts are part of the parameter name (that is, not an index). In terms of letter choices, variables and parameters for commodity and factor *quantities* start with the letter *q;* for commodity and factor *prices*, the first letters are *p* and *w*, respectively.

In GAMS, indices are called sets.

Sets

$a \in A$	activities {AGR-A agricultural activity NAGR-A nonagricultural activity}
$c \in C$	commodities {AGR-C agricultural commodity NAGR-C nonagricultural commodity}
$f \in F$	factors {LAB labor CAP capital}
$h \in H$	households {U-HHD urban household R-HHD rural household}

Parameters

ad_a	efficiency parameter in the production function for activity a
cpi	consumer price index (CPI)
$cwts_c$	weight of commodity c in the CPI
$shry_{hf}$	share for household h in the income of factor f
qfs_f	supply of factor f
α_{fa}	share of value-added for factor f in activity a
β_{ch}	share in household h consumption spending of commodity c
θ_{ac}	yield of output c per unit of activity a

Variables

P_c	market price of commodity c
PA_a	price of activity a
Q_c	output level in commodity c
QA_a	level of activity a
QF_{fa}	demand for factor f from activity a
QH_{ch}	consumption of commodity c by household h
YF_{hf}	income of household h from factor f
WF_f	price of factor f
YH_h	income of household h

Equations. Production and commodity block

Activity Production Function

$$QA_a = ad_a \cdot \prod_{f \in F} QF_{fa}^{\alpha_{fa}} \qquad a \in A \tag{1}$$

Factor Demand

$$WF_f = \frac{\alpha_{fa} \cdot PA_a \cdot QA_a}{QF_{fa}} \qquad f \in F, a \in A \tag{2}$$

Activity Price

$$PA_a = \sum_{c \in C} \theta_{ac} \cdot P_c \qquad a \in A \tag{3}$$

Commodity Output

$$Q_c = \sum_{a \in A} \theta_{ac} \cdot QA_a \qquad c \in C \tag{4}$$

Institution block

Factor Income

$$YF_{hf} = shry_{hf} \cdot WF_f \cdot \sum_{a \in A} QF_{fa} \qquad h \in H, f \in F \tag{5}$$

Household Income

$$YH_h = \sum_{f \in F} YF_{hf} \qquad h \in H \tag{6}$$

Household Demand

$$QH_{ch} = \frac{\beta_{ch} \cdot YH_h}{P_c} \qquad c \in C, h \in H \tag{7}$$

System constraint block

Factor Market Equilibrium

$$\sum_{a \in A} QF_{fa} = qfs_f \qquad f \in F \tag{8}$$

Output Market Equilibrium

$$Q_c = \sum_{h \in H} QH_{ch} \qquad c \in C \tag{9}$$

Price Normalization Equation

$$\sum_{c \in C} cwts_c \cdot P_c = cpi \tag{10}$$

Data base

Table 2—Social accounting matrix for Exercise 1

	AGR-A	NAGR-A	AGR-C	NAGR-C	LAB	CAP	U-HHD	R-HHD	TOTAL
AGR-A			125						125
NAGR-A				150					150
AGR-C							50	75	125
NAGR-C							100	50	150
LAB	62	55							117
CAP	63	95							158
U-HHD					60	90			150
R-HHD					57	68			125
TOTAL	125	150	125	150	117	158	150	125	

Solve the model in GAMS and verify that the solution can replicate the above SAM (Table 2).

3. Practical advises and coding of the model

You may run the input file in GAMS at any point in the process of constructing the model. If the model is incomplete and hence not solved, GAMS will nevertheless check that the input conforms with its syntax, report any errors, and, in the absence of errors, carry out other instructions, including displays. To catch errors at an early stage, it is often helpful to inspect the results of displays of elements (sets, variables, and parameters), that have been defined via operations.

Sets correspond exactly to the indices in the algebraic representation of the model. In order to facilitate SAM-related computations, it is helpful to generate a global set, here named AC, including all elements in the sets for factors, activities, commodities, and households. Sets for the latter items are subsequently declared and defined as subsets of the global set.

When building CGE models, it is often useful to have identical sets with different names. In GAMS, the ALIAS command may be used to create a set that is identical to a set that already has been defined. (In the suggested answer, ALIAS is used to define sets identical to AC, C, and F.)

```
*PARAMETERS=====================================================
```

SETS

AC global set (SAM accounts and other items)

/AGR-A agricultural activity

NAGR-A non-agricultural activity

AGR-C agricultural commodity

NAGR-C non-agricultural commodity

LAB labor

CAP capital

U-HHD urban household

R-HHD rural household

TOTAL total account in SAM /

The following example is used to illustrate the assignment of membership to dynamic sets.

set item all items / dish,ink,lipstick,pen,pencil,perfume /
subitem1(item) first subset of item / pen,pencil /
subitem2(item) second subset of item;

subitem1("ink") = yes ; subitem1("lipstick") = yes ;
subitem2(item) = yes ; subitem2("perfume") = no ;
display subitem1, subitem2;

Note that the sets subitem1 and subitem2 are declared like any other set. The two sets become dynamic because of assignments. They are also domain checked: the only members they will ever be able to have must also be members of item. And item is a static set and henceforth its membership is frozen. The first two assignments each add one new element to subitem1. The third is an example of the familiar indexed assignment: subitem2 is assigned all the members of item. The output caused by the display statement, that will reveal the membership of the sets, is shown below for verification.

---- *7 SET SUBITEM1 first subset of item*
INK , LIPSTICK, PEN , PENCIL

---- *7 SET SUBITEM2 second subset of item*

DISH , INK , LIPSTICK, PEN , PENCIL

ACNT(AC) all elements in AC except total

ACNT(AC) = YES;ACNT("TOTAL") = NO;

A(AC) activities

/AGR-A, NAGR-A/

C(AC) commodities

/AGR-C, NAGR-C/

F(AC) factors

/LAB, CAP/

H(AC) households

/U-HHD, R-HHD/ ;

The alias statement is used to give another name to a previously declared set.

ALIAS(AC,ACP); ALIAS(C,CP); ALIAS(F,FP);

ALIAS(ACNT,ACNTP);

*PARAMETERS===

PARAMETERS

ad(A) efficiency parameter in the production fn for a

alpha(F,A) share of value-added to factor f in activity a

beta(C,H) share of household consumption spending on commodity c

cpi consumer price index

cwts(C) weight of commodity c in the CPI

qfs(F) supply of factor f

shry(H,F) share for household h in the income of factor f

theta(A,C) yield of output c per unit of activity a

;

*VARIABLES===

Variables mean free variable, with an allowed range going from minus infinity to plus infinity. In chapters 3 and 4 we met positive variables, with range going from 0 to plus infinity.

VARIABLES

P(C) price of commodity c

PA(A) price of activity a

Q(C) output level for commodity c

QA(A) level of activity a

QF(F,A) quantity demanded of factor f from activity a

QH(C,H) quantity consumed of commodity c by household h

WF(F) price of factor f

YF(H,F) income of household h from factor f

YH(H) income of household h

;

*EQUATIONS===

In this model we write equations, not inequations. Thus, there are no complementary variables associated to the equations.

EQUATIONS

*PRODUCTION AND COMMODITY BLOCK

PRODFN(A) Cobb-Douglas production function for activity a

FACDEM(F,A) demand for factor f from activity a

OUTPUTFN(C) output of commodity c

PADEF(A) price for activity a

*INSTITUTION BLOCK

FACTTRNS(H,F) transfer of income from factor f to h-hold h

HHDINC(H) income of household h

HHDEM(C,H) consumption demand for household h & commodity c

```
*SYSTEM CONSTRAINT BLOCK
        FACTEQ(F) market equilibrium condition for factor f
        COMEQ(C) market equilibrium condition for commodity c
        PNORM price normalization
 ;
*PRODUCTION AND COMMODITY BLOCK++++++++

PRODFN(A).. QA(A) =E= ad(A)*PROD(F, QF(F,A)**alpha(F,A));

FACDEM(F,A).. WF(F) =E= alpha(F,A)*PA(A)*QA(A) / QF(F,A);

OUTPUTFN(C).. Q(C) =E= SUM(A, theta(A,C)*QA(A));

PADEF(A).. PA(A) =E= SUM(C, theta(A,C)*P(C));
```

The summation notation in GAMS can be used for simple and complex expressions. The format is based on the idea of always thinking of a summation as an operator with two arguments: ***Sum(index of summation, summand)*** *A comma separates the two arguments, and if the first argument requires a comma then it should be in parentheses. The second argument can be any mathematical expression including another summation.*

As a simple example, the transportation problem contains the expression

Sum(j, x(i,j))

that is equivalent to $\sum_j x_{ij}$

A slightly more complex summation is used in the following example:

Sum((i,j), c(i,j)*x(i,j))

that is equivalent to $\sum_i \sum_j x_{ij}$

The last expression could also have been written as a nested summation as follows:

Sum(i, Sum(j, c(i,j)*x(i,j)))

Products are dfined in GAMS using exactly the same format as summations, replacing Sum by Prod. For example,

prod(j, x(i, j))

is equivalent to: $\prod_j x_{ij}$

```
*INSTITUTION BLOCK+++++++++++++++++++++

FACTTRNS(H,F).. YF(H,F) =E= shry(H,F)*WF(F)*SUM(A, QF(F,A));

HHDINC(H).. YH(H) =E= SUM(F, YF(H,F));

HHDEM(C,H).. QH(C,H) =E= beta(C,H)*YH(H)/P(C);



*SYSTEM CONSTRAINT BLOCK+++++++++++++++

FACTEQ(F).. SUM(A, QF(F,A)) =E= qfs(F);

COMEQ("AGR-C").. Q("AGR-C") =E= SUM(H, QH("AGR-C",H));

 PNORM.. SUM(C, cwts(C)*P(C)) =E= cpi;
```

If you have implemented the model correctly, there should be no "significant" discrepancies between left-hand and right-hand sides of the equations when GAMS plugs in your parameter values and initial variable levels. You check this by looking for three asterisks (***) in the "equation listing." If the left-hand side value (indicated as "LHS = <value>") is "significantly" different (let's say by more than 1.0E-5) from the right-hand side value in the preceding equation (after =E=), a problem exists with the definitions of the parameters and variables that appear in the equation in question.

```
*MODEL=========================================================
MODEL
        CGE1 Simple CGE model /ALL/
 ;
```

If all previously defined equations are to be included, you can enter /all/ in place of the explicit list.
Do not input the row and column totals of the SAM. Instead, to remove one source of errors and model infeasibility, compute row and column totals and check that they are identical.

Assume that base-year factor and output prices are at unity and assign parameter and variable quantities on this basis. (This amounts to choosing the unit for each real flow so as to assure that its corresponding price is one.)
Table is a new element of The SAMLS' code.

*SOCIAL ACCOUNTING MATRIX=======================================
It is vary useful to have the table format for data entry. In chapter 3 and 4 we entered data by lists (the values given to the parameters)

TABLE SAM(AC,ACP) social accounting matrix

	AGR-A	NAGR-A	AGR-C	NAGR-C	LAB	CAP	U-HHD	R-HHD
AGR-A			125					
NAGR-A				150				
AGR-C							50	75
NAGR-C							100	50
LAB	62	55						
CAP	63	95						
U-HHD					60	90		
R-HHD					57	68		

;

PARAMETER
tdiff(AC) column minus row total for account ac;
*This parameter is used to check that the above SAM is balanced.
SAM("TOTAL",ACNTP) = SUM(ACNT, SAM(ACNT,ACNTP));
SAM(ACNT,"TOTAL") = SUM(ACNTP, SAM(ACNT,ACNTP));
tdiff(ACNT) = SAM("TOTAL",ACNT)-SAM(ACNT,"TOTAL");
Here we have an example of data entry by direct assignment
DISPLAY SAM, tdiff;

*ASSIGNMENTS FOR PARAMETERS AND VARIABLES=======================

PARAMETERS
*The following parameters are used to define initial values of
*model variables.
Now we calibrate the model

P0(C), PA0(A), Q0(C), QA0(A), QF0(F,A), QH0(C,H), WF0(F), YF0(H,F),
YH0(H)
;

*PRODUCTION AND COMMODITY BLOCK++++++++

P0(C) = 1;
PA0(A) = 1;
WF0(F) = 1;
*prices normed to 1

Q0(C) = SAM("TOTAL",C)/P0(C);
QA0(A) = SAM("TOTAL",A)/PA0(A);
QF0(F,A) = SAM(F,A)/WF0(F);
* initial values given by the SAM

alpha(F,A) = SAM(F,A) / SUM(FP, SAM(FP,A));
ad(A) = QA0(A) / PROD(F, QF0(F,A)**alpha(F,A));
theta(A,C) = (SAM(A,C)/P0(C)) / QA0(A);
* values of the parameters

*INSTITUTION BLOCK++++++++++++++++++++

QH0(C,H) = SAM(C,H)/P0(C);
YF0(H,F) = SAM(H,F);
YH0(H) = SAM("TOTAL",H);

```
beta(C,H) = SAM(C,H)/SUM(CP, SAM(CP,H));
shry(H,F) = SAM(H,F)/SAM("TOTAL",F);

*SYSTEM CONSTRAINT BLOCK+++++++++++++++

cwts(C) = SUM(H, SAM(C,H)) / SUM((CP,H), SAM(CP,H));
cpi = SUM(C, cwts(C)*P0(C));
qfs(F) = SAM(F,"TOTAL")/WF0(F);

*INITIALIZING ALL VARIABLES++++++++++++

P.L(C) = P0(C);
PA.L(A) = PA0(A);
Q.L(C) = Q0(C);
QA.L(A) = QA0(A);
QF.L(F,A) = QF0(F,A);
QH.L(C,H) = QH0(C,H);
YF.L(H,F) = YF0(H,F);
WF.L(F) = WF0(F);
YH.L(H) = YH0(H);
* P.L is the variable, P0 is a parameter computed before and uses as an initial guess
* of the value of the variable
*DISPLAY++++++++++++++++++++++++++++++++

DISPLAY
ad, alpha, beta, cpi, cwts, qfs, shry, theta,
* display the values of he parameters

P.L, PA.L, Q.L, QA.L, QF.L, QH.L, WF.L, YF.L, YH.L
* initial guesses
;

*SOLVE STATEMENT FOR BASE========================================
```

*SOLVE CGE1 USING MCP;

A value of unity for all factor and commodity prices (that were initialised at this level) is a reliable indicator that the initial model solution replicates the initial equilibrium as captured by the initial SAM. Actually, the model should be able to reproduce the SAM

After having solved the model we can read the results. We can also transform and display them with GAMS statements.

```
*REPORT SETUP AND BASE REPORT====================================

*SET AND PARAMETERS FOR REPORTS++++++++

SET
SIM simulations
/BASE base simulation
CINCR increase in capital stock/
* BASE reproduces the SAM; CINCR is a variant simulation

PARAMETERS

QFSCAPSIM(SIM) capital supply for sim'on sim (experiment parameter)
*Parameter is used to change the value for the capital stock
*parameter before solving the model for simulation sim

QFSREP(F,SIM) supply of factor f for simulation sim (value used)
PREP(C,SIM) demander price for commodity c
PAREP(A,SIM) price of activity a
QREP(C,SIM) output level for commodity c
QAREP(A,SIM) level of activity a
QFREP(F,A,SIM) demand for factor f from activity a
QHREP(C,H,SIM) consumption of commodity c by household h
WFREP(F,SIM) price of factor f
YFREP(H,F,SIM) income of household h from factor f
```

YHREP(H,SIM) income of household h

SAMREP(SIM,AC,ACP) SAM computed from model solution

BALCHK(AC,SIM) column minus row total for account ac in SAM

*these parameters give the values of the main variables for each of the two simulations. They

*will be displayed

;

QFSCAPSIM("BASE") = qfs("CAP");

*supply of capital in base (the SAM)

QFSCAPSIM("CINCR") = 1.1*qfs("CAP");

* supply of capital in the variant

DISPLAY QFSCAPSIM;

In addition to model statement and base solution, the GAMS input file with the suggested answer includes a LOOP where two simulations are carried out: the BASE and a simulation for which the capital stock is increased by 10 percent.

The loop statement causes GAMS to execute the statements within the scope of the loop for each member of the driving set(s) in turn.

Consider a hypothetical case when a growth rate is empirical:

set t / 1985*1990 /

parameter pop(t) / 1985 3456 /

growth(t) / 1985 25.3, 1986 27.3, 1987 26.2

1988 27.1, 1989 26.6, 1990 26.6 /;

The loop statement is then used to calculate the cumulative sums

loop(t, pop(t+1) = pop(t) + growth(t)) ;

in an iterative rather than a parallel way. In this example there is one statement in the scope of the loop, and one driving, or controlling, set.

```
LOOP(SIM,

        qfs("CAP") = QFSCAPSIM(SIM);

        SOLVE CGE1 USING MCP;

        QFSREP(F,SIM) = qfs(F);

        PREP(C,SIM) = P.L(C);
        PAREP(A,SIM) = PA.L(A);
        QREP(C,SIM) = Q.L(C);
        QAREP(A,SIM) = QA.L(A);
        QFREP(F,A,SIM) = QF.L(F,A);
        QHREP(C,H,SIM) = QH.L(C,H);
        WFREP(F,SIM) = WF.L(F);
        YFREP(H,F,SIM) = YF.L(H,F);
        YHREP(H,SIM) = YH.L(H);
* computation of the parameters, which will be displayed

*Payments from activities
        SAMREP(SIM,F,A) = WF.L(F)*QF.L(F,A);
*Payments from commodities
        SAMREP(SIM,A,C) = P.L(C)*theta(A,C)*QA.L(A);
*Payments from factors
        SAMREP(SIM,H,F) = YF.L(H,F);
*Payments from households
        SAMREP(SIM,C,H) = P.L(C)*QH.L(C,H);

);
```

A set of report parameters are created. For each model solution they show (1) values for the factor supply parameter (which was changed in the experiment QFSREP); (2) solution values for all model variables; (3) a SAM that is defined using data from each model solution. The

SAM parameters provide data on the budgets of all agents and markets in the model in a concise format.

```
*Computing totals for SAMREP
SAMREP(SIM,"TOTAL",ACNTP) = SUM(ACNT, SAMREP(SIM,ACNT,ACNTP));
SAMREP(SIM,ACNT,"TOTAL") = SUM(ACNTP, SAMREP(SIM,ACNT,ACNTP));

*Check that SAMREP is balanced
BALCHK(ACNT,SIM) = SAMREP(SIM,"TOTAL",ACNT)-
SAMREP(SIM,ACNT,"TOTAL");



OPTION QFREP:3:1:1, QHREP:3:1:1, YFREP:3:1:1, SAMREP:3:1:1;
DISPLAY
QFSREP, PREP, PAREP, QREP, QAREP, QFREP, QHREP, WFREP, YFREP, YHREP,
SAMREP, BALCHK
* display the results of the two simulations
;
```

4. Output of the simulation

Echo prints (lines 5 to 324)

Whether or not errors prevent your optimization problem from being solved, the first section of output from a GAMS run is an echo, or copy, of your input file. For the sake of future reference, GAMS puts line numbers on the left-hand side of the echo.

Display (lines 332 to 464)

Gives the list of all the parameters and variables of the model, with their numerical values.

FIRST SIMULATION BASE

Equations listing (lines 438 to 543 and 587 to 705)

A product of the solve command, the equation listing shows the specific instance of the model that is created when the current values of the sets and parameters are plugged into the general algebraic form of the model. For example, the generic demand constraint given in the input file for the transportation model is

demand(j) .. sum(i, x(i,j)) =g= b(j) ;

while the equation listing of specific constraints is

--------demand =g= satisfy demand at market j
demand(new-york).. x(seattle, new-york) +x(san-diego, new-york) =g= 325 ;
demand(chicago).. x(seattle, chicago) +x(san-diego, chicago) =g= 300 ;
demand(topeka).. x(seattle, topeka) +x(san-diego, topeka) =g= 275 ;

The default output also contains a section called the column listing, analogous to the equation listing, which shows the coefficients of specific variables for each generic variable. This listing would be particularly useful for verifying a GAMS model that was previously implemented in MPS format.

Model statistics (lines 709 to 727)

It is a group of statistics about the model's size.

MODEL STATISTICS

```
BLOCKS OF EQUATIONS          10     SINGLE EQUATIONS          24
BLOCKS OF VARIABLES           9     SINGLE VARIABLES          24
NON ZERO ELEMENTS            73     NON LINEAR N-Z            36
DERIVATIVE POOL               9     CONSTANT POOL             30
CODE LENGTH                 245


GENERATION TIME      =        0.078 SECONDS      4 Mb  WIN230-230 Feb 12, 2009
```

The BLOCK counts refer to the number of generic equations and variables. The SINGLE counts refer to individual rows and columns in the specific model instance being generated. For nonlinear models, some other statistics are given to describe the degree of non-linearity in the problem.

Solution report (lines 730 to 778)

After the solver executes, GAMS prints out a brief *solve summary* whose two most important entries are `SOLVER STATUS` and the `MODEL STATUS`.

Solution Report SOLVE CGE1 Using MCP From line 244

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

   MODEL   CGE1
   TYPE    MCP
   SOLVER  PATH              FROM LINE  244

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL

 RESOURCE USAGE, LIMIT          0.064      1000.000
 ITERATION COUNT, LIMIT         0         10000
 EVALUATION ERRORS              0             0
```

The desired solver status is 1 NORMAL COMPLETION

General reports (lines 782 to 854, 856 to 923 and 926 to 930)

Fists, we have the value of each equation. Most equations were written as f(x)=0? Then, its value is zero. Then, we have the values of the variables. At the end of the solvers solution report is a very important report summary, which gives a tally of the total number of non-optimal, infeasible, and unbounded rows and columns.

**** REPORT SUMMARY : 0 NONOPT
0 INFEASIBLE
0 UNBOUNDED
0 REDEFINED
0 ERRORS

THEN WE HAVE

SECOND SIMULATION CINCR: the supply of capital is increased by 10 per cent

- Equation listing: lines 933 to 1040
- Column listing: lines 1044 to 1202
- Model statistics: lines 1212 to 1275
- Value of each equation: lines 1279 to 1351
- Values of variables: lines 1353 to 1420
- Report summary: lines 1423 to 1427

Finally we have the display of the parameters in the loop from QFSREP to BALCHK: lines 1433 to 1600.