A Guide to the GAMS-input-file

This is a user's guide to the GAMS-input-file of the regional CGE model described throughout section 3. It transforms the mathematical program specified in section 4.1 into an executable computer program based in GAMS (the acronym stands for General Algebraic Modeling System). To make this chapter self-contained we reproduce some introductory material on the construction of a CGE-model in GAMS, however we recommend the reading of GAMS tutorial (Brooke et al., Chap. 2).

We provide the complete GAMS-PROGRAM in this link {click here to download CRS2.GMS}. It’s clearly only a prototype and the numerical values of the parameters and initial values were explained in section 3.2. For a collection of models with similar specification, but somehow more sophisticated, we offer the following link: {Oklahoma State University; Department of Agricultural Economics; RCGE}

In what follows, the GAMS-Program input file is presented and explained in its major components.

Index sets

The application starts with a definition of the main index sets and subsets. A set declaration consists of declaring and specifying the index to be used. Sets should be declared before their subsets. Every declaration consists of a logical name, a label field, followed by a list of elements of the index set. As such, it is the same as indexes used in the equations of the model. They correspond to the subscription notation of table 4.2{click here to go to table 4.2}.

```
$TITLE REGIONAL CGE MODEL FOR OKLAHOMA (1993)(CRS.GMS)

$OFFSYMLIST OFFSYMXREF OFFUPPER

SETS

i       Sectors            /Agr  agriculture

                             Min  mining

                             Man  manufacture

                          SER services/

ag(i)   Agricultural sectors            / AGR/

nag(i)  Nonagricultural market sectors   / MIN, SER, MAN/

f       Factors          /L       labor,  K      capital,  T
land/

fl(f)   Factors not land / L, K/

ALIAS(i,j);
```

A $-sign at the beginning of the program, is used for special commands, i.e., $TITLE, where we introduce the title of the model. All GAMS-statements end with a semicolon. The ALIAS-statement defines an alternative name for an index set (subscript).

BASE YEAR DATA

Base year variables are based upon the Social Accounting Matrix (SAM) and are distinguished by "0" as a suffix in their names, i.e., L0(i) states base year labor. GAMS requires a DECLARATION and ASSIGNMENT of each variable or parameter. Here, we declare the base year variables as parameters. GAMS offers flexible arrangements for introducing the parameters (variables). We recommend first to declare (initiate) all the parameters, then use tables to enter data and finally, assign the values.

To provide better readability, parameters are declared by blocks: prices, production, income and expenditure blocks. In GAMS, comment-lines and text in general are introduce by "*" in the first column of a row.

```
*#####-- DECLARATION OF BASE YEAR VARIABLES (AS PARAMENTERS)

PARAMETERS

*@Price block
  PL0           Wage rate
  PLROC0        Wage rate of rest-of-country
  PKROC0        Cap rate of rest-of-country
  PK0(i)        cap rate
  PT0(ag)       Land rent
  PE0(i)        Export price
  PM0(i)        Import price
  PR0(i)        Reg price
  P0(i)         Composite price
  PN0             Net output price or value-added price of
sector i
  PX0(i)     Composite price face for producers

*@Production block
  L0(i)         Labor demand
  LS0           Labor supply by hh
  TLS0          Total labor supply
  LHHH0         Labor employed by household group
  LGOV0         Labor employed by gov
  K0(i)         capital demand
  T0(i)         Land demand
  KS0           Supply of pri capital
  TKS0          Total pri capital supply
  TS0           Supply of land
  VA0(i)        Value added
  V0(j,i)       Composite intermediate good demand
  TV0(i)        Composite intermediate good total demand
  VR0(j,i)      Reg int good demand
  VM0(j,i)      Imported int good demand
  TVR0(i)       Reg int good total demand
  TVM0(i)       Imported int good total demand
  IBT0(I)    Indirect business taxes
  X0(i)         Sector output
```

```
        E0(i)          Export of reg product
        M0(i)          Import
        R0(i)          Reg supply of reg product

      *@Income block
        LY0            Labor income
        KY0            capital income
        TY0            Land income
        YENT0          Gross Enterprise income
        YH0            Household income
        DYH0           Disposable hh income
        HSAV0          Household saving
        SAV0           Total saving
        ROWSAV0        Saving from rest-of-world
        TRGOV0         Gov transfer to hh
        REMIT0             Remittance from outside the region to
household
        YGOV0          Gov revenue
        ENTY0          Enterprise income distrib to hhs
        GOVITR0        Inter gov transfer
        GOVBOR0        Government Borrowing
        GRP0           Gross regional product

      *@Expenditure block
        HEXP0          Household expend
        QR0(i)         Demand for reg consump good
        QM0(i)         Demand for imp consump good
        Q0(i)          Demand for comp consump good
        GOVEXP0        government expenditure
        QGOVR0(i)      government demand for reg good
        QGOVM0(i)      government demand for imported good
        QGOV0(i)       government demand for comp good
        QInvR0(i)      Invest demand for reg good
        QInvM0(i)      Invest demand for imported good
        QInv0(i)       Invest demand for comp good
        INV0           Total invest
```

The following variables are defined as "logical variables". A logical variable takes the value of 1 if the condition stated is true and "0" if not. We use these variables when defining an equation or for assigning value to a particular variable depending on the "true" or "false" condition of a specific condition, i.e., variable NZV takes the value of “1” if both regional and imported intermediate input are used, according to the following graph.

```
************************************
*Regional         x     x     0     0     0=zero, x=not zero
*Import           x     0     x     0
*
*NZV              T     F     F     F     T=TRUE,  F=FALSE
*ZVR              F     F     T     F
*ZVM              F     T     F     T
************************************
```

```
        ZVM(i,J)  non imported intermediate demand with-or-without
regional interm. demand
        ZVR(i,J)  only imported intermediate demand
        NZV(i,J)  both imported intermediate demand and regional
demand

        ZQM(i)    non imported final demand and either none or
some regional final demand for household
```

ZQR(i)   only imported final demand for household
NZQ(i)   both imported final demand and regional final demand for households

ZGOVM(i)
ZGOVR(i)
NZGOV(i)

ZInvM(i)
ZInvR(i)
NZInv(i)

DECLARATION OF PARAMETERS TO BE CALIBRATED

These parameters are those specified in Table 4.5. {Click here to see table 4.5}. They are declared in the following segment of the application but they will be initialized later.

```
*#####-- DECLARATION OF PARAMETERS TO BE CALIBRATED.

PARAMETERS

*This parameters are those specified in Table 5.5.

*@Production block
  a0(i)              composite value added req per unit of output i
  a(j,i)          req of interm good j per unit of good i
  Alpha(i,f)      value added share param
  Ava(i)          value added shift param
  RHOv(i)         interm input subs param
  deltav1(j,i)
  deltav(j,i)     interm input share param
  Av(j,i)         interm input shift param
  RHOx(i)         output transformation param
  deltax1(i)
  deltax(i)       output share param
  Ax(i)           output shift param

*@Income block
  ktax            capital tax rate
  sstax           factor income tax rate for labor
  ttax            factor income tax rate for land
  retr            rate of retained earnings fr ent inc
  et          enterprise tax rate
  hhtax           income tax rate for hh
  ltr             Household Income Transfer Coefficient
  mps             saving rate
  ibtax(i)        indirect business tax
  beta(i)         param calc fr elast of comm demand wrt inc

*@Expenditure block
  RHOq            consumer demand subs param
  deltaq1(i)
  deltaq(i)       consumer demand share param
  Aq(i)           consumer demand constant eff param
  RHOgov          gov demand subs param
  deltagov1
  deltagov        gov demand share param
  Agov            gov demand constant eff param
  RHOinv          inv gov demand subs param
```

```
deltainv1
deltainv       inv gov demand share param
Ainv           inv gov demand constant eff param
;
```

DATA

Data comes from our SAM (Table 2.1). {Click here to see table 2.1} You should note that values from our SAM are scaled to millions of dollars instead of thousands. Though the scaling of our data is not a "must" for solving the model, we strongly recommend scaling. Scaling problems have been found to create more serious problems in more disaggregated models.

```
Table      IOR(i,j) Input-output regional matrix

AGR   MIN          MAN        SER
AGR   675.798     8.115       863.991         34.800
MIN   123.47      2180.942    1258.117    881.343
MAN   159.671     1390.701    3594.97         3953.2
SER   381.542     1317.332    5272.186    9752.027


;
Table    IOM(i,j)   Input-output import matrix

      AGR   MIN          MAN    SER
AGR   579.870     5.160 378.422     41.300
MIN   11.850      1274.869    311.094     385.272
MAN   446.830     450.977     8835.472    2750.345
SER   155.160     458.802     1886.710    4188.764

;

Table VAD(i,f)  Value added matrix
           L           K              T
AGR        433.242    571.360        709.066
MIN       1622.806    2713.109
MAN       7577.427    4025.159
SER      20767.388    12042.708
;


Table HHCONR(i,*)  Household consumption demand for regional
goods

        HOUSE
AGR     147.210
MIN    1587.998
MAN    2656.085
SER    30727.366
;

Table HHCONM(i,*)  Household consumption demand for imported
goods
        HOUSE
AGR    181.550
MIN    141.662
MAN   5713.705
SER   9510.103
;

Table  GOVCONR(i,*)    Government  consumption  demand  for
regional goods
```

```
                 GOV
      AGR      12.863
      MIN     231.250
      MAN    1854.066
      SER    1477.995
      ;

      Table   GOVCONM(i,*)   Government   consumption   demand   for
imported goods
                 GOV
      AGR     20.097
      MIN     29.912
      MAN    823.846
      SER    542.893
      ;

      Table FYDIST(*,f)  Factor income distribution to households
                        L          K             T
      HH           31363.057          0.00          683.300
      ;

      TABLE ParamA(*,i)   BASE YEAR VALUES FOR INDUSTRY
                          AGR                     MIN               MAN
SER

      PT0                    1.00              1.00              1.00
1.00
      PK0                    1.00              1.00              1.00
1.00
      PR0                    1.00              1.00              1.00
1.00
      P0                     1.00              1.00              1.00
1.00
      PM0                    1.00              1.00              1.00
1.00
      PE0                    1.00              1.00              1.00
1.00
      X0                   4344.160        12089.784       34190.427
59115.190
      R0               1752.557       6282.217       18360.150
49486.101
      E0               2591.603       5807.567       15830.277
9629.089
      M0               1216.846      2170.418         21475.978
16920.731
      IBT0              96.301  666.971          186.879
4318.043
      QINVR0            9.780     19.097         4751.457
557.653
      QINVM0            10.447    15.759         2454.803
178.299
      SIGMAp              1.00001          1.00001          1.00001
1.00001
      SIGMAv              1.42              0.5               3.55
4
      SIGMAx               3.90              2.90              2.90
0.70
      SIGMAq               1.42              0.50              3.55
2.00
      SIGMAgov            1.42              0.50              3.55
2.00
```

```
      SIGMAinv           1.42             0.50             3.55
2.00

      ;

      TABLE ParamB(f,*)   BASE YEAR VALUES FOR FACTORS
           WAGE0   WAGEROC0     FTAX0         RETENT0       CAP0
CAPROC0
      L     1.0       1.0     6126.715           0
      K                       -1006.686      9077.096           1
1
      T                        25.766            0
      ;

      TABLE ParamC (*,*)   BASE YEAR VALUES FOR HOUSEHOLD GROUPS

                   HTAX0          HSAV0          TRGOV0
      HOUSE        6976.571    -3869.320       11490.516

      +           REMIT0        ENTYDIS0
      HOUSE       760.824       9582.303
      ;

      TABLE ParamD(g,*)   BASE YEAR VALUES FOR GOVTS
                   BOR0         GOVDR0    GOVDM0
      GOV            0.0       3576.174   1416.748
      ;
      SCALAR LHHH0      Labor used by household /  107.070/;
      SCALAR LGOV0      Labor used by government      /6981.839/;
      SCALAR GOVITR0    Inter-government transfers    /8477.813/;
      SCALAR YENT0      Enterprise income             /20359.022/;
      SCALAR ENTTAX0    Enterprise taxes              /1699.623/;
      SCALAR ROWGOV0    Rest  of  world  transfers  to  government
/4375.094/;
      SCALAR ROWSAV0    Saving from ROW               /2789.519/;
      SCALAR QINVMSUM0  Investment  demand  for  imported  goods  /
2659.308/;
      SCALAR etaL       Labor migration elasticity     /      .92
/;
      SCALAR etaK       Capital migration elasticity   /      .92
/;
      Scalar KMobil     Capital Mobility              /        1.0
/;
```

ASSIGNING VALUES: Initialization of Parameters

Here, we assign a value to each of the base year variables declared previously. This assigning of values should correspond to our SAM.

```
      *@Production block
        L0(i)        =VAD(i,"L");
        K0(i)        =VAD(i,"K");
        T0(i)        =VAD(i,"T");
        VA0(i)       =sum(f,VAD(i,f));
        V0(j,i)      =IOR(j,i)+IOM(j,i);
        TV0(i)       =sum(j,V0(i,j));
        VM0(j,i)     =IOM(j,i);
        VR0(j,i)     =IOR(j,i);
        TVM0(i)      =sum(j,VM0(i,j));
        TVR0(i)      =sum(j,VR0(i,j));
        LHHH0        =LHHH0;
```

```
        LGOV0          =LGOV0;
        LS0            =sum(i,VAD(i,"L"))+LHHH0+LGOV0;
        X0(i)          =ParamA("X0",i);
        E0(i)          =ParamA("E0",i);
        R0(i)          =ParamA("R0",i);
        KS0(i)         =VAD(i,"K");
        TKS0           =sum(i,KS0(i));
        TS0(i)         =VAD(i,"T");
        IBT0(I)        =PARAMA("IBT0",I);

      *@Income block
        TRGOV0         =ParamC ("HOUSE","TRGOV0");
        LY0            =sum(i,VAD(i,"L"))+LHHH0+LGOV0;
        KY0            =sum(i,VAD(i,"K"));
        TY0            =sum(i,VAD(i,"T"));
        YENT0          =YENT0;
        REMIT0         =ParamC ("HOUSE","REMIT0");
        YH0
=sum(f,FYDIST("HH",f))+ParamC("HOUSE","ENTYDis0")+TRGOV0
                        +REMIT0;
        DYH0           =YH0 -ParamC ("HOUSE","HTAX0");
        HSAV0          =ParamC ("HOUSE","HSAV0");
        HEXP0          =DYH0-HSAV0-LHHH0;
        SAV0                         =ParamB("K","RETENT0")+   ParamC
("HOUSE","HSAV0")+ROWSAV0;
        ROWSAV0        =ROWSAV0;
        YGOV0
=sum(i,ParamA("IBT0",i))+sum(f,ParamB(f,"FTAX0"))

+ParamC("HOUSE","HTAX0")+ENTTAX0+ROWGOV0+GOVITR0;
        ENTY0          =ParamC("HOUSE","ENTYDis0");
        GOVBOR0         =ParamD("GOV","BOR0");
        GRP0           =LY0+KY0+TY0+sum(i,ParamA("IBT0",i));

      *@Expenditure block
        QR0(i)         =HHCONR(i,"HOUSE");
        QM0(i)         =HHCONM(i,"HOUSE");
        Q0(i)          =QM0(i)+QR0(i);
        GOVEXP0
=ParamD("GOV","GOVDR0")+ParamD("GOV","GOVDM0")
                        +ParamC("HOUSE","TRGOV0")+LGOV0+GOVITR0;
        QGOVR0(i)      =GOVCONR(i,"GOV");
        QGOVM0(i)      =GOVCONM(i,"GOV");
        QGOV0(i)       =QGOVM0(i)+QGOVR0(i);
        QINVR0(i)      =ParamA("QINVR0",i);
        QINVM0(i)      =ParamA("QINVM0",i);
        QINV0(i)       =QINVM0(i)+QINVR0(i);
        INV0           =sum(i,QINV0(i));
        M0(i)          =ParamA("M0",i);

      *@Price block
        PL0            =ParamB("L","WAGE0");
        PK0(i)         =ParamA("PK0",i);
        PLROC0         =ParamB("L","WAGEROC0");
        PKROC0         =ParamB("K","CAPROC0");
        PT0(ag)         =ParamA("PT0",ag);
        PE0(i)         =ParamA("PE0",i);
        PM0(i)         =ParamA("PM0",i);
        PR0(i)        =ParamA("PR0",i);
        P0(i)          =ParamA("P0",i);
        PX0(i)        =(PR0(i)*R0(i)+PM0(i)*M0(i))/(R0(i)+M0(i));

      *---------------------------------------------------
      * Regional     x  x  0  0     0=zero, x=not zero
```

```
* Import       x  0  x  0
*
* NZV          T  F  F  F     T=True, F=False
* ZVR          F  F  T  F
* ZVM          F  T  F  T
*-----------------------------------------------------

  ZVM(i,j)     =(VM0(i,j) eq 0);
  ZVR(i,j)     =(VR0(i,j) eq 0) and (VM0(i,j) ne 0);
  NZV(i,j)     =(VR0(i,j) ne 0) and (VM0(i,j) ne 0);

  ZQM(i)    =(QM0(i) eq 0);
  ZQR(i)    =(QR0(i) eq 0) and (QM0(i) ne 0);
  NZQ(i)    =(QR0(i) ne 0) and (QM0(i) ne 0);

  ZGOVM(i)      =(QGOVM0(i) eq 0);
  ZGOVR(i)      =(QGOVR0(i) eq 0) and (QGOVM0(i) ne 0);
  NZGOV(i)      =(QGOVR0(i) ne 0) and (QGOVM0(i) ne 0);

  ZInvM(i)     =(QInvM0(i) eq 0);
  ZInvR(i)     =(QInvR0(i) eq 0) and (QInvM0(i) ne 0);
  NZInv(i)     =(QInvR0(i) ne 0) and (QInvM0(i) ne 0);
```

So far, we have assigned values to our base year variables (parameters). Has GAMS read the assignments correctly? Next, we define new parameter to check for accuracy of our assignment statements. If correct, we should get our SAM and a block of unity prices. Though, the DISPLAY statement of GAMS allows the modeler to easily see the assignment results with statements like

```
DISPLAY PK0, PT0, L0, K0,TSO;
```

we prefer to define new parameters, so the output will be easier to read and presented in table format. The advantage of this procedure may not be appreciated in small CGE models, but definitely are greatly appreciated in much bigger models.

```
PARAMETER SAM1 SOCIAL ACOUNTING MATRIX -BASE YEAR PRICES-;
SAM1(I,"PK")=PK0(I);
SAM1(ag,"PT")=PT0(ag);
SAM1(I,"PE0")=PE0(I);
SAM1(I,"PM0")=PM0(I);
SAM1(I,"PR0")=PR0(I);
SAM1(I,"P0")=P0(I);
SAM1(I,"PR0")=PR0(I);

PARAMETER SAM2  SOCIAL ACCOUNTING MATRIX -BASE YEAR DATA-;
SAM2(I,"L0")=L0(I);
SAM2(I,"K0")=K0(I);
SAM2(I,"KS0")=KS0(I);
SAM2(I,"T0")=T0(I);
SAM2(I,"TS0")=TS0(I);
SAM2(I,"VA0")=VA0(I);
SAM2(I,"TVR0")=TVR0(I);
SAM2(I,"TVM0")=TVM0(I);
SAM2(I,"TV0")=TV0(I);
SAM2(I,"IBT0")=IBT0(I);
SAM2(I,"X0")=X0(I);
SAM2(I,"M0")=M0(I);
SAM2(I,"R0")=R0(I);
SAM2(I,"E0")=E0(I);
SAM2(I,"Q0")=Q0(I);
SAM2(I,"QR0")=QR0(I);
SAM2(I,"QM0")=QM0(I);
SAM2(I,"QGOV0")=QGOV0(I);
```

```
SAM2(I,"QGOVR0")=QGOVR0(I);
SAM2(I,"QGOVM0")=QGOVM0(I);
SAM2(I,"QINV0")=QINV0(I);
SAM2(I,"QINVR0")=QINVR0(I);
SAM2(I,"QINVM0")=QINVM0(I);

OPTION DECIMALS=0;
DISPLAY SAM1;

OPTION DECIMALS=3;
DISPLAY SAM2;

DISPLAY V0,VM0,VR0,LS0,PL0, PLROC0,LHHH0,LGOV0,LY0,KY0,TY0,
YENT0,REMIT0,YH0,DYH0,YGOV0,GRP0,HSAV0,HEXP0,GOVEXP0,SAV0,RO
WSAV0,
TRGOV0,ENTY0,ENTTAX0,GOVBOR0;
```

PARAMETER CALIBRATION

Calibration is the setting of model parameters in order to make the equilibrium solution fit the data of a given base year (our SAM). The way to perform this adjustment in GAMS is to solve at fixed (consistent) values of observed variables, treating some of the parameters as variables. The solution will then fit the model to the data.

The calibration procedure was introduced in section 2.3. We have linked the text equation that is used in the calibration with each of our definitions; i.e., clicking over the definition

```
a0(i)   = VA0(i)/X0(i);
```

takes you to equation 3.1.2 in our text. Once again, remember that our base year variables are identified by a “0” suffix in the name.

```
*###########################################################*
*                                                           *
*                   PARAMETER CALIBRATION                   *
*                                                           *
*###########################################################*


*#####-- CALIBRATION

*@Production block

  a0(i)           =VA0(i)/X0(i);
  a(j,i)          =V0(j,i)/X0(i);
  alpha(ag,"K")   =VAD(ag,"K")/VA0(ag);
  alpha(ag,"T")   =VAD(ag,"T")/VA0(ag);
  alpha(ag,"L")   =1-alpha(ag,"K")-alpha(ag,"T");
  alpha(nag,"K")  =VAD(nag,"K")/VA0(nag);
  alpha(nag,"L")  =1-alpha(nag,"K");
  Ava(ag)         =VA0(ag)/Prod(f,VAD(ag,f)**alpha(ag,f));
  Ava(nag)
=VA0(nag)/PROD(fl,VAD(nag,fl)**alpha(nag,fl));
  RHOv(i)         =1-1/ParamA("SIGMAv",i);
  deltav1(j,i)
    $(NZV(j,i))                    =(VR0(j,i)/VM0(j,i))**(1-
RHOv(j))*(PR0(j)/PM0(j));
  deltav(j,i)
    $(NZV(j,i))   =1/(1+deltav1(j,i));
```

```
        Av(j,i)
          $(NZV(j,i))   =V0(j,i)/(deltav(j,i)*VM0(j,i)**RHOv(j)
                         +(1-deltav(j,i))
                         *VR0(j,i)**RHOv(j))**(1/RHOv(j));
        RHOx(i)        =1+1/ParamA("SIGMAx",i);
        deltax1(i)                              =(R0(i)/E0(i))**(1-
RHOx(i))*(PR0(i)/PE0(i));
        deltax(i)      =1/(1+deltax1(i));
        Ax(i)                 =X0(i)/(deltax(i)*E0(i)**RHOx(i)+(1-
deltax(i))
                         *R0(i)**RHOx(i))**(1/RHOx(i));

      *@Income block
        sstax         =ParamB("L","FTAX0")/LY0;
        ktax          =ParamB("K","FTAX0")/KY0;
        ttax          =ParamB("T","FTAX0")/TY0;
        retr          =ParamB("K","RETENT0")/sum(i,VAD(i,"K"));
        ibtax(i)      =ParamA("IBT0",i)/(PR0(i)*X0(i));
        et        =ENTTAX0/KY0;
        hhtax         =ParamC("HOUSE","HTAX0")/YH0 ;
        ltr           =1;
        mps           =ParamC("HOUSE","HSAV0")/YH0 ;

      *@Expenditure block

        RHOq(i) = 1-1/ParamA("SIGMAq",i);
        deltaq1(i)$NZQ(i)          =          (QR0(i)/QM0(i))**(1-
RHOq(i))*(PR0(i)/PM0(i));
        deltaq(i)$NZQ(i) =1/(1+deltaq1(i));
        Aq(i)$NzQ(i)     =     Q0(i)/(deltaq(i)*QM0(i)**RHOq(i)+(1-
deltaq(i))*QR0(i)**RHOq(i))**(1/RHOq(i));
        RHOgov(i) = 1-1/ParamA("SIGMAgov",i);
        deltagov1(i)$NZGOV(i)          =(QGOVR0(i)/QGOVM0(i))**(1-
RHOgov(i))*(PR0(i)/PM0(i));
        deltagov(i)$NZGOV(i) = 1/(1+deltagov1(i));
        Agov(i)$NZGOV(i)                                          =
QGOV0(i)/(deltagov(i)*QGOVM0(i)**RHOgov(i)+(1-
deltagov(i))*QGOVR0(i)**RHOgov(i))**(1/RHOgov(i));
        RHOinv(i)= 1-1/ParamA("SIGMAinv",i);
        deltainv1(i)$NZInv(i)     =     (QINVR0(i)/QINVM0(i))**(1-
RHOinv(i))*(PR0(i)/PM0(i));
        deltainv(i)$NZInv(i) = 1/(1+deltainv1(i));
        Ainv(i)$NZInv(i)                                          =
QINV0(i)/(deltainv(i)*QINVM0(i)**RHOinv(i)+(1-
deltainv(i))*QINVR0(i)**RHOinv(i))**(1/RHOinv(i));
        beta(i) = Q0(i)*P0(i)/HEXP0;
```

To check values for the calibration we use the following parameters which allow us to display the results of calibration in a table-like display.

```
PARAMETER CALIBR PARAMETER CALIBRATED;
CALIBR(I,"A0")=A0(I);
CALIBR(I,"AVA")=AVA(I);
CALIBR(I,"RHOV")=RHOv(I);
CALIBR(I,"RHOQ")=RHOQ(I);
CALIBR(I,"DELTAQ")=DELTAQ(I);
CALIBR(I,"AQ")=AQ(I);
CALIBR(I,"IBTAX")=IBTAX(I);
CALIBR(I,"RHOGOV")=RHOGOV(I);
CALIBR(I,"DELTAGOV")=DELTAGOV(I);
CALIBR(I,"AGOV")=AGOV(I);
CALIBR(I,"RHOINV")=RHOINV(I);
```

```
CALIBR(I,"AINV")=AINV(i);
CALIBR(I,"RHOX")=RHOX(i);
CALIBR(I,"DELTAX")=DELTAX(i);
CALIBR(I,"AX")=AX(i);
CALIBR(I,"BETA")=BETA(i);
DISPLAY CALIBR;
DISPLAY a,Av,deltav,alpha,
ktax,sstax,ttax,retr,et,mps,hhtax;
```

VARIABLE DECLARATION

All symbols belonging to the list of choice variables in the mathematical program should be declared as VARIABLES, not as PARAMETERS. Endogenous variables are given in table 4.3. {Click here to see table 4.3} Every endogenous variable declaration has a logical name followed by a label field (optional).

```
*#########################################################*
*                                                         *
*      VARIABLE DECLARATION                               *
*                                                         *
*#########################################################*

* ENDOGENOUS VARIABLES

VARIABLES
  Z             Objective Function Value

*@Price block
  PL                Wage rate
  PK(i)             Capital rate
  PKL                Capital rate in the long run
  PT(ag)            Land rent
  PN(i)             Net price
  PR(i)             Regional price
  P(i)              Composite price
  PX(i)             Composite price faced by consumers

*@Production block
  LAB(i)            Labor demand
  CAP(i)            Capital demand
  LAND(ag)           Land demand
  TCAP              Total Capital Demand
  TLAB              Total Labor Demand
  LS                Labor supply
  LMIG              Labor migration
  KMIG              Capital migration
  VA(i)             Value added
  V(j,i)            Composite intermediate good demand
  VM(j,i)           Imported int good demand
  VR(j,i)           Reg int good demand
  R(i)              Regional supply
  X(i)              Output
  EXP(i)            Export
  M(i)              Import
  TVM(i)            Imported int good total demand
  TVR(i)            Reg int good total demand
  TV(i)             Composite intermediate good total demand
  adjL              Labor adjustment

*@Income block
```

```
        LY                  Labor income (original hhs)
        ALY                    Adjusted labor income (staying + in-
migrating)
        KY                  capital income (original capital stock)
        TY                  Land income
        YENT                Enterprise income
        RETENT              Retained Earnings by enterprises
        YH                      Income of hh staying in the region
(including in-migrants)
        DYH                    Disposable hh income (staying in the
region + inmigra)
        HSAV                Household saving (staying +inmigrat)
        SAV                 Total saving
        INV                 Investment
        YGOV                gov revenue
        IBTX                Indirect business tax
        GRP                 Gross region product

      *### Expenditure block
        AHEXP                  Adjusted household expenditure (spent
within the region)
        Q(i)                Demand for comp consump good
        QM(i)               Demand for imp consump good
        QR(i)               Demand for reg consump good
        GOVEXP              gov expend
        QGOV(i)             gov demand for comp good
        QGOVM(i)            gov demand for imported good
        QGOVR(i)            gov demand for reg good
        QINV(i)             Invest gov demand for comp good
        QINVM(i)            Invest gov demand for imported good
        QINVR(i)            Invest gov demand for reg good
        SLACK(i)
        SLACK2(i)
```

The following statement ensures that we are working with positive variables. All variables may be assigned as positive variables except the "Z" variable which we use in the optimization statement.

```
      POSITIVE VARIABLE   SLACK, SLACK2;
```

Equation Declaration

This section declares the equations of the model which are those presented in Table 4.1. {Click here for table 4.1} Equations are also denoted by symbols. Hence, every equation can be referred to by its logical name.

```
*###########################################################*
*                                                           *
*      EQUATION DECLARATION                                 *
*                                                           *
*###########################################################*

*This section declares the equations of the model
*which are those presented in table 5.1

EQUATIONS
  EQZ                 objective function

*@Price block
  NETprice(i)       net price
  Price(i)          composite price
  Price1(i)
```

```
*@Production block
  Ldemand(i)       labor demand
  KdemandSR(i)       capital demand
  KdemandLR(i)
  Tdemand(ag)       land demand
  TLdem             total labor demand
  TKdem             total capital demand
  VAdemand(i)       value added demand
  Vdemand(j,i)      intermediate demand
  VAprod1(nag)      value added prod fc
  VAprod2(ag)       value added prod fc
  Vces(j,i)         ces fc for int demand
  TVdemand(i)       intermediate total demand
  TVRdemand(i)      int reg total demand
  TVMdemand(i)      int imp total demand
  VRdem(j,i)        demand for reg int good
  VRdem0(j,i)        demand for reg int good for goods with
zero import
  VMDem0(j,i)        demand for imp int good for goods with
zero import
  Xcet(i)           cet fc for reg product
  Rsupply(i)        reg supply of reg product
  LSupply           labor supply
  LMIGrat           labor migration
  adjustL           labor migration adjustment
  KMIGrat           capital migration
  KMIGrat1


*@Income block
  LYincome          labor income
  ALYincome         adjusted labor income
  KYincomeSR          capital income
  KYincomeLR
  TYincome          land income
  YENTincome        enterprise income
  RETearn           Retained earning by enterprises
  YHincome          household income
  DHYincome         disposable income
  HSAVings          household savings
  SAVings           total savings
  INVest            total investment
  YGOVincome        Government income
  INDtax            Indirect business tax
  GRProduct         gross region product

*@Expenditure block
  AHEXPLow          adj. household expenditure
  Qces              ces fc for consumption
  Qdemand           cons demand for composite good
  QRdem0            cons demand for reg goods
  QRdem1
  QRdem2
  QMdem1
  QMdem2
  GOVEXPend         Gov expenditure
  QGOVces           ces for st and loc gov demand
  QGOVdemand        st and loc gov cons
  QGOVRdem0         st and loc gov reg cons
  QGOVRDem1
  QGOVRDem2
  QGOVMDem1
  QGOVMDem2
```

```
  QINVces          ces for invest gov demand
  QINVemand        invest gov cons
  QINVRdem0        invest gov reg cons
  QInvRdem1
  QInvRdem2
  QInvMdem1
  QInvMdem2
  Mimports(i)     import

*@Equilibrium
  COMMequil(i)     comm market equilibrium
  Lequil           labor market equilibrium
  Kequil(i)        cap market equilibrium
  Kequil1
  Tequil(ag)        land market equilibrium;
```

EQUATION DEFINITION

All equations are defined following the algebraic specification given in Table 4.1. {Click here to see table 4.1.} This section requires special attention and intense scrutiny. To help the reader, we have linked each equation definition. Thus, it is possible to move from GAMS-specification format to its algebraic specification. Furthermore, each algebraic equation in Table 4.1 is itself linked to the part of the text where derivation takes place.

In the equation definition, the “=E=” represents an equality-sign; the greater-or-equal sign is written as =G=, and smaller-or-equal as =L=. Thus, by comparing with Table 5.1 algebraic specification, the meaning of each equation is straightforward. Exceptions are equations involving a dollar expression; i.e., QCES(CI)$NZQ(CI). A dollar expression indicates that the value of the variable (i.e., the equation QCES) should be considered only if the expression that follows is true.

```
*###########################################################*
      *                                                           *
      *      EQUATION DEFINITION                                  *
      *                                                           *
      *###########################################################*

      *All equations are defined following the algebraic structure
      *on table 5.1.

        EQZ..           Z       =e= sum(i,SLACK(i)+SLACK2(i));

      *@Price block
        NETprice(i)..     PN(i)     =e=  PX(i)-sum(j,A(j,i)*P(j))-
ibtax(i)*PX(i);
        Price(i)..                          P(i)             =e=
(PR(i)*R(i)+PM0(i)*M(i))/(R(i)+M(i));
        Price1(i)..                          PX(i)           =e=
(PR(i)*R(i)+PE0(i)*Exp(i))/(R(i)+Exp(i));

      *@Production block
        Ldemand(i)..    LAB(i) =e= alpha(i,"L") *PN(i)*X(i)/PL;
        KdemandSR(i)$(Not    Kmobil)..             CAP(i)    =e=
alpha(i,"K")*PN(i)*X(i)/PK(i);
        KdemandLR(i)$(Kmobil)..                   CAP(i)     =e=
alpha(i,"K")*PN(i)*X(i)/PKL;
        Tdemand(ag)..              LAND(ag)=e=     alpha(ag,"T")
*PN(ag)*X(ag)/PT(ag);
```

```
        TLdem..          TLAB  =e= Sum(i,LAB(i));
        TKdem..          TCAP  =e= Sum(i,CAP(i));
        LSupply ..       LS    =e= LS0;
        LMIGrat ..       LMIG  =e= etaL*LS0*LOG(PL/PLROC0);
        adjustL..        adjL  =e= (LS0+LMIg)/LS0;
        KMIGrat$(KMobil)..                                    KMIG
=e=etaK*(SUM(i,K0(i))*LOG(PKL/PKROC0));
        KMIGrat1$(not KMobil).. KMIG       =e= 0;
        VAdemand(i)..     VA(i)+SLACK(i)+SLACK2(i)=e= a0(i)*X(i);
        VAprod1(nag)..                VA(nag)                 =e=
Ava(nag)*LAB(nag)**alpha(nag,"L")*CAP(nag)**
                                     alpha(nag,"K");
        VAprod2(ag)..               VA(ag)                    =e=
Ava(ag)*LAB(ag)**alpha(ag,"L")*CAP(ag)**

alpha(ag,"K")*LAND(ag)**alpha(ag,"T");
        Vdemand(j,i)..    V(j,i)  =e= a(j,i)*X(i);
        Vces(j,i)..       V(j,i)  =e= Av(j,i)*(deltav(j,i)*VM(j,i)
                                    **RHOv(j)+(1-deltav(j,i))

*VR(j,i)**RHOv(j))**(1/RHOv(j));
        TVdemand(i)..     TV(i)    =e= sum(j,V(i,j));
        VRdem(j,i)$NZV(j,i)..
                          VR(j,i) =e= VM(j,i)*((1-deltav(j,i))/
                                      deltav(j,i)*
                                      PM0(j)/PR(j))**(1/(1-
RHOv(j)));
        VRdem0(j,i)$ZVM(j,i)..   VR(j,i) =e= V(j,i);
        VMdem0(j,i)$ZVM(j,i)..   VM(j,i) =e= 0;
        TVRdemand(i)..    TVR(i)  =e= sum(j,VR(i,j));
        TVMdemand(i)..    TVM(i)  =e= sum(j,VM(i,j));
        Xcet(i)..                         X(i)                =e=
Ax(i)*(deltax(i)*EXP(i)**RHOx(i)+(1-
                                    deltax(i))*R(i)**RHOx(i))
                                    **(1/RHOx(i));
        Rsupply(i)..             R(i)           =e=   EXP(i)*((1-
DELTAx(i))/DELTAx(i)
                                    *PE0(i)/PR(i))**(1/(1-
RHOx(i)));
        INDtax..          IBTX    =E= Sum(i,ibtax(i)*X(i));
        GRProduct..       GRP     =e= ALY + KY + TY + IBTX;


      *@Income block

      *ALY is defined for all labor; LY is defined for original
household
        ALYincome..      ALY      =e=  PL*(TLAB+LHHH0+LGOV0);
        LYincome..        LY        =e=  ALY+PLROC0*(SQRT(LMig**2)-
LMig)*0.5
                                     -
PL*(SQRT(LMig**2)+LMig)*0.5;
        KYincomeSR$(not kmobil)..   KY   =e= sum(i,PK(i)*CAP(i));
        KYincomeLR$(kmobil)..             KY                  =e=
sum(i,PKL*CAP(i))+PKROC0*(SQRT(KMIG**2)-KMIG)
                                         *0.5-
PKL*(SQRT(KMIG**2)+KMIG)*0.5;
        RETearn..        RETENT  =e= retr*KY;
        TYincome..       TY      =e= sum(ag,PT(ag)*LAND(ag));
        YENTincome..     YENT    =e= KY*(1-ktax);
        YHincome ..      YH      =e= ALY*(1-sstax)
                                   +TY*(1-ttax)+(YENT-RETENT-
et*KY)
                                   +REMIT0+adjL*TRGOV0
```

```
                                    -((SQRT((adjL-1)**2)-(adjL-
1))*0.5)
                                    *(TY*(1-ttax)+(YENT-RETENT-
et*KY)
                                    +REMIT0);
        DHYincome ..   DYH   =e= YH *(1-hhtax );
        HSAVings ..    HSAV  =e= mps *YH ;
        SAVings..        SAV     =e= HSAV+RETENT+ROWSAV0;
        INVest..         INV     =e= sum(i,P(i)*QINV(i));
        YGOVincome..     YGOV     =e= Sum(i,ibtax(i)*PX(i)*X(i))
                                     +sstax*ALY
                                     +ktax*KY+et*KY
                                     +ttax*TY
                                     +hhtax *YH+GOVBOR0+GOVITR0;
      *@Expenditure block
        AHEXPLow..   AHEXP  =e= DYH-HSAV-PL*LHHH0;
        Qdemand(i)..  Q(i)   =e= beta(i)*AHEXP/P(i);
        Qces(i)$NZQ(i).. Q(i)  =e= Aq(i)*(deltaq(i)*QM(i)
                     **RHOq(i)+(1-deltaq(i))*QR(i)**RHOq(i))
                                      **(1/RHOq(i));
        QRdem0(i)$NZQ(i)..       QR(i)       =e=       QM(i)*((1-
deltaq(i))/deltaq(i)
                                  *PM0(i)/PR(i))**(1/(1-RHOq(i)));
        QRdem1(i)$ZQM(i).. QM(i) =e= 0;
        QMdem1(i)$ZQM(i).. QR(i) =e= Q(i);
        QRdem2(i)$ZQR(i).. QR(i) =e= 0;
        QMdem2(i)$ZQR(i).. QM(i) =e= Q(i);
        GOVEXPend..                    GOVEXP               =e=
sum(i,P(i)*QGOV(i))+adjL*
                                        TRGOV0+PL*LGOV0+GOVITR0;
        QGOVdemand(i)..   QGOV(i)   =e= QGOV0(i);
        QGOVces(i)$NZGOV(i)..                 QGOV(i)       =e=
Agov(i)*(deltagov(i)
                                    *QGOVM(i)**RHOgov(i)+(1-
deltagov(i))

*QGOVR(i)**RHOgov(i))**(1/RHOgov(i));
        QGOVRdem0(i)$NZGOV(i)..       QGOVR(i)   =e=QGOVM(i)*((1-
deltagov(i))
                          /deltagov(i)*PM0(i)/PR(i))**(1/(1-
RHOgov(i)));
        QGOVRdem1(i)$ZGOVM(i)..   QGOVM(i) =e= 0;
        QGOVMdem1(i)$ZGOVM(i)..   QGOVR(i) =e= QGOV(i);
        QGOVRdem2(i)$ZGOVR(i)..   QGOVR(i) =e= 0;
        QGOVMdem2(i)$ZGOVR(i)..   QGOVM(i) =e= QGOV(i);
        QINVemand(i)..   QINV(i)  =e= QINV0(i);
        QINVces(i)$NZInv(i)..                            QINV(i)
=e=Ainv(i)*(deltainv(i)*QINVM(i)
                                        **RHOinv(i)+(1-
deltainv(i))*QINVR(i)**RHOinv(i))
                                        **(1/RHOinv(i));
        QINVRdem0(i)$NZInv(i)..     QINVR(i)=e=     QINVM(i)*((1-
deltainv(i))
                                  /deltainv(i)*PM0(i)/PR(i))**(1/(1-
RHOinv(i)));
        QInvRDem1(i)$ZInvM(i).. QInvM(i)=e= 0;
        QInvMDem1(i)$ZInvM(i).. QInvR(i)=e= QInv(i);
        QInvRDem2(i)$ZInvR(i).. QInvR(i)=e= 0;
        QInvMDem2(i)$ZInvR(i).. QInvM(i)=e= QInv(i);
        Mimports(i)..               M(i)                   =e=
TVM(i)+QM(i)+QGOVM(i)+QINVM(i);

      *@Equilibrium
```

```
        COMMequil(i)..
X(i)+M(i)=e=TV(i)+Q(i)+QGOV(i)+QINV(i)+EXP(i);
        Lequil..          sum(i,LAB(i))+LHHH0+LGOV0 =e= LS0+LMIG;
        Kequil1$(KMobil)..        KMig =e= Sum(i,CAP(i)-KS0(i));
        Kequil(i)$(not KMobil).. CAP(i)          =e= KS0(i);
        Tequil(ag)..      LAND(ag)  =e= T0(ag);
```

STARTING VALUES and BOUNDS

Before a model is solved, it is necessary to initialize all relevant bounds. Bounds are treated in the same way as parameters. Here, we introduce GAMS language to characterize a variable. A GAMS-variable is characterized by a suffix:

- .L current level of the variable
- .M shadow price on the bound
- .LO lower bound
- .UP upper bound
- .FX fixed (lower bound=upper bound).

The variables (.L-values) keep their ***level*** value from one solution to the next assignment. Unassigned upper bounds are set at plus infinity, non-initialized lower bounds at minus infinity. In direct assignments, variables should be referenced with their suffices. The initialization is at arbitrary values, in order to test the computational procedure. However, in empirical applications it is recommended to initialize the variables at their SAM-values.

```
      *##########################################################*
      *                                                          *
      *      INITIALIZATION OR STARTING VALUES                   *
      *                                                          *
      *##########################################################*
      *@Price block                     *@Income block
        PL.L        =PL0        ;
        PKL.L        =1;
        PK.L(i)   =PK0(i)      ;
        PT.L(ag)      =PT0(ag)      ;              HSAV.L    =HSAV0
;
        PR.L(i)     =PR0(i)     ;              YGOV.L        =YGOV0
;
        P.L(i)     =P0(i)      ;
        PX.L(i)   = PX0(i)    ;
        PN.L(i)      = PX0(i)-sum(j,A(j,i)*P0(j))-ibtax(i)*PX0(i);

      *@Production block
        SLACK.L(i) =0;              SLACK2.L(i)   =0;
        LAB.L(i)  =L0(i)    ;     INV.L      =INV0;
        CAP.L(i) =K0(i)   ;       GRP.L      =GRP0;
      *                             *@Expenditure block
        LAND.L(ag) =T0(ag)    ;

        LS.L     = LS0;
        LMIG.L   =0;
        KMIG.L    =0;
        VA.L(i)   =VA0(i)   ;
        VM.L(j,i)  =VM0(j,i)  ;
        VR.L(j,i)  =VR0(j,i)  ;          QM.L(i) =QM0(i)   ;
        V.L(j,i)   =V0(j,i)   ;
        TVM.L(i)   =TVM0(i)   ;
        TVR.L(i)    =TVR0(i)    ;            GOVEXP.L     =GOVEXP0
;
        TV.L(i)    =TV0(i)    ;          QGOV.L(i)  =QGOV0(i)    ;
        R.L(i)    =R0(i)    ;          QGOVM.L(i) =QGOVM0(i)   ;
```

```
  X.L(i)     =X0(i)     ;          QGOVR.L(i) =QGOVR0(i)   ;
  EXP.L(i)  =E0(i)    ;
  M.L(i)    =M0(i)    ;
  Q.L(i)     =beta(i)*HEXP0/PX0(i);
  QR.L(i) =QR0(i)   ;

*@Income block
  LY.L       =LY0       ;
  KY.L      =KY0        ;
  TY.L       =TY0       ;
  adjL.L      =1     ;
  YENT.L     =YENT0     ;
  YH.L     =YH0     ;
  SAV.L      =SAV0      ;
  DYH.L    =DYH0    ;
  QINVM.L(i) =QINVM0(i) ;
  QINVR.L(i) =QINVR0(i) ;
  QINV.L(i)  =QINV0(i)  ;

*###########################################################*
*                                                           *
*      VARIABLE BOUNDS                                      *
*                                                           *
*###########################################################*

  PL.LO       = 0.000001;
  PT.LO(ag)    = 0.000001;
  PK.LO(i)   =  0.000001;
  PR.LO(i)    = 0.000001;
  PN.LO(i)   = 0.000001;
  P.LO(i)     = 0.000001;
  R.LO(i)    = 0.000001;
  PX.LO(i)   =  0.000001;
  QM.LO(i)$(QM0(i) ne 0)  = 0.000001;
  QR.LO(i)$(QR0(i) ne 0)  = 0.000001;
  Q.LO(i)$(Q0(i) ne 0)    = 0.000001;
  QM.LO(i)$(QM0(i) eq 0)  = 0;
  QR.LO(i)$(QR0(i) eq 0)  = 0;
  Q.LO(i)$(Q0(i) eq 0)    = 0;
  VR.LO(i,j)$(VR0(i,j) ne 0)  = 0.000001;
  VM.LO(i,j)$(VM0(i,j) ne 0)  = 0.000001;
  V.LO(i,j)$(V0(i,j) ne 0)    = 0.000001;
  VR.LO(i,j)$(VR0(i,j) eq 0)  = 0;
  VM.LO(i,j)$(VM0(i,j) eq 0)  = 0;
  V.LO(i,j)$(V0(i,j) eq 0)    = 0;
```

The follow statement uses GAMS-Options to reduce the amount of output and computer time assigned to solve the model. This is not recommended for beginners who may do better by getting more output from GAMS. Especially, for those having problems obtaining a “zero error message”. Iterlim, limrow, lincol and solprint, will limit the number of iterations, suppress the printing of equations, suppress the printing of columns, and suppress the list of the solution, respectively. Although this saves paper, we do not recommend it unless you understand your model very well and have your model running without error messages.

```
OPTIONS ITERLIM=5000, LIMROW=0, LIMCOL=0, SOLPRINT=OFF;
```

MODEL and SOLVE statements

A group of equations constitute a mathematical model. GAMS uses the statement **MODEL** to allow us to specify which equations should be considered as part of our mathematical model. In addition, we need to give a name to our model, i.e.; our Model is called OKLAHOMA.

```
*-- MODEL DEFINITION AND SOLVE STATEMENT

MODEL OKLAHOMA /ALL/;
```

We use in our example all the declared equations, if that would not be the case, instead of the word “ALL” we would have written each equation needed.

Equilibrium is found by minimizing the objective function ***EQZ*** that calculates the absolute sum of deviations (Slack variables). This process was introduced in section 4.2. {Click here to review section 4.2} The GAMS-statement to solve the mathematical program defined by the model OKLAHOMA with objective Z, using the MINOS5 non-linear programming algorithm NLP, reads as:

```
SOLVE OKLAHOMA MINIMIZING Z USING NLP;
```

REPORTING VALIDATION OF THE MODEL

When an equilibrium solution has been computed, the results are sorted in tabulation format. We define tables for commodity balances, prices, consumer budgets, etc.. These tables give the level of the endogenous variables of OKLAHOMA model. If they are correct, the values of these tables validate with those of our base year (SAM values). We call this process the validation of the model.

```
*-- SOLUTION DISPLAY STATEMENT
*-- SOLUTION VALUES OF ENDOGENOUS VARIABLES


PARAMETER VALID VARIABLES FOR THE VALIDATION OF THE MODEL;
VALID(i,"SLACK1") = SLACK.L(i);
VALID(i,"SLACK2") = SLACK2.L(i);
VALID(i,"PR") = PR.L(i);
VALID(i,"P") = P.L(i);
VALID(i,"PN") = PN.L(i);
VALID(i,"PK") = PK.L(i);
VALID(ag,"PT") = PT.L(ag);
VALID(i,"PX") = PX.L(i);
VALID(i,"PE") = PE0(i);
VALID(i,"X") = X.L(i);
VALID(i,"R") = R.L(i);
VALID(i,"EXP") =EXP.L(i);
VALID(i,"M") = M.L(i);
VALID(i,"VA") = VA.L(i);
VALID(i,"LAB") =LAB.L(i);
VALID(i,"CAP") =CAP.L(i);
VALID(ag,"LAND") =LAND.L(ag);
VALID(i,"TVR") =TVR.L(i);
VALID(i,"TVM") =TVM.L(i);
VALID(i,"TV") =TV.L(i);
VALID(i,"Q") =Q.L(i);
VALID(i,"QR") =QR.L(i);
```

```
VALID(i,"QM") =QM.L(i);
VALID(i,"QGOV") =QGOV.L(i);
VALID(i,"QGOVR") =QGOVR.L(i);
VALID(i,"QGOVM") =QGOVM.L(i);
VALID(i,"QINV") =QINV.L(i);
VALID(i,"QINVR") =QINVR.L(i);
VALID(i,"QINVM") =QINVM.L(i);

PARAMETER VALID2  -INTERMEDIATE USE MATRIX-;
VALID2(I,"AGR","V")=V.L(I,"AGR");
VALID2(I,"MIN","V")=V.L(I,"MIN");
VALID2(I,"MAN","V")=V.L(I,"MAN");
VALID2(I,"SER","V")=V.L(I,"SER");
VALID2(I,"AGR","VR")=VR.L(I,"AGR");
VALID2(I,"MIN","VR")=VR.L(I,"MIN");
VALID2(I,"MAN","VR")=VR.L(I,"MAN");
VALID2(I,"SER","VR")=VR.L(I,"SER");
VALID2(I,"AGR","VM")=VM.L(I,"AGR");
VALID2(I,"MIN","VM")=VM.L(I,"MIN");
VALID2(I,"MAN","VM")=VM.L(I,"MAN");
VALID2(I,"SER","VM")=VM.L(I,"SER");

PARAMETER VALID3 -VALIDATION OF THE MODEL-;
VALID3("OBJECTIVE") = Z.L;
VALID3("PL") = PL.L;
VALID3("LMIG")=LMIG.L;
VALID3("KMIG")=KMIG.L;
VALID3("TCAP")=TCAP.L;
VALID3("TLAB")=TLAB.L;
VALID3("LS")=LS.L;
VALID3("LMIG")=LMIG.L;
VALID3("ADJL") = ADJL.L;
VALID3("LY")=LY.L;
VALID3("ALY")=ALY.L;
VALID3("KY")=KY.L;
VALID3("TY")=TY.L;
VALID3("YENT") = YENT.L;
VALID3("RETENT")=RETENT.L;
VALID3("YH")=YH.L;
VALID3("PL") = PL.L;
VALID3("DYH")=DYH.L;
VALID3("HSAV")=HSAV.L;
VALID3("SAV")=SAV.L;
VALID3("INV") = INV.L;
VALID3("YGOV")=YGOV.L;
VALID3("GOVEXP")=GOVEXP.L;
VALID3("IBTX")=IBTX.L;
VALID3("GRP")=GRP.L;
VALID3("AHEMP")=AHEXP.L;

option decimals=3;
DISPLAY VALID,VALID2,VALID3;
```

SIMULATION

Before starting a simulation run, one should specify the name of the scenario (here, simul1). The last step in preparing the model is to define the index sets and parameters for reporting. We define post-equilibrium variables that we use in constructing indexes for the relevant variables.

```
*######## SIMULATION ############*
```

```
PE0(i)=1.1;
model simul1 /all/;
solve simul1 minimizing z using nlp;
OPTION SOLPRINT=OFF;

*-- SOLUTION DISPLAY STATEMENT
*-- SOLUTION VALUES OF ENDOGENOUS VARIABLES


PARAMETER PRICES MARKET CLEARING PRICES;
PRICES(i,"SLACK1") = SLACK.L(i);
PRICES(i,"SLACK2") = SLACK2.L(i);
PRICES(i,"PR") = PR.L(i);
PRICES(i,"P") = P.L(i);
PRICES(i,"PN") = PN.L(i);
PRICES(i,"PK") = PK.L(i);
PRICES(ag,"PT") = PT.L(ag);
PRICES(i,"PX") = PX.L(i);
PRICES(i,"PE") = PE0(i);

PARAMETER PROD1 MARKET CLEARING PRODUCTION VARIABLES;
PROD1(i,"X") = X.L(i);
PROD1(i,"R") = R.L(i);
PROD1(i,"EXP") =EXP.L(i);
PROD1(i,"M") = M.L(i);
PROD1(i,"VA") = VA.L(i);
PROD1(i,"LAB") =LAB.L(i);
PROD1(i,"CAP") =CAP.L(i);
PROD1(ag,"LAND") =LAND.L(ag);

PARAMETER TRADE1 MARKET CLEARING PRODUCTION VARIABLES;
TRADE1(i,"TVR") =TVR.L(i);
TRADE1(i,"TVM") =TVM.L(i);
TRADE1(i,"TV") =TV.L(i);
TRADE1(i,"Q") =Q.L(i);
TRADE1(i,"QR") =QR.L(i);
TRADE1(i,"QM") =QM.L(i);
TRADE1(i,"QGOV") =QGOV.L(i);
TRADE1(i,"QGOVR") =QGOVR.L(i);
TRADE1(i,"QGOVM") =QGOVM.L(i);
TRADE1(i,"QINV") =QINV.L(i);
TRADE1(i,"QINVR") =QINVR.L(i);
TRADE1(i,"QINVM") =QINVM.L(i);

PARAMETER PRODUCT2  -PRODUCTION SYSTEMS VARIABLES-;
PRODUCT2(I,"AGR","V")=V.L(I,"AGR");
PRODUCT2(I,"MIN","V")=V.L(I,"MIN");
PRODUCT2(I,"MAN","V")=V.L(I,"MAN");
PRODUCT2(I,"SER","V")=V.L(I,"SER");
PRODUCT2(I,"AGR","VR")=VR.L(I,"AGR");
PRODUCT2(I,"MIN","VR")=VR.L(I,"MIN");
PRODUCT2(I,"MAN","VR")=VR.L(I,"MAN");
PRODUCT2(I,"SER","VR")=VR.L(I,"SER");
PRODUCT2(I,"AGR","VM")=VM.L(I,"AGR");
PRODUCT2(I,"MIN","VM")=VM.L(I,"MIN");
PRODUCT2(I,"MAN","VM")=VM.L(I,"MAN");
PRODUCT2(I,"SER","VM")=VM.L(I,"SER");

PARAMETER OTHER1 MARKET CLEARING VALEUES OF VARIABLES;
OTHER1("OBJECTIVE") = Z.L;
OTHER1("PL") = PL.L;
OTHER1("LMIG")=LMIG.L;
OTHER1("KMIG")=KMIG.L;
OTHER1("TCAP")=TCAP.L;
```

```
OTHER1("TLAB")=TLAB.L;
OTHER1("LS")=LS.L;
OTHER1("LMIG")=LMIG.L;
OTHER1("ADJL") = ADJL.L;
OTHER1("LY")=LY.L;
OTHER1("ALY")=ALY.L;
OTHER1("KY")=KY.L;
OTHER1("TY")=TY.L;
OTHER1("YENT") = YENT.L;
OTHER1("RETENT")=RETENT.L;
OTHER1("YH")=YH.L;
OTHER1("PL") = PL.L;
OTHER1("DYH")=DYH.L;
OTHER1("HSAV")=HSAV.L;
OTHER1("SAV")=SAV.L;
OTHER1("INV") = INV.L;
OTHER1("YGOV")=YGOV.L;
OTHER1("GOVEXP")=GOVEXP.L;
OTHER1("IBTX")=IBTX.L;
OTHER1("GRP")=GRP.L;
OTHER1("AHEMP")=AHEXP.L;

option decimals=3;

DISPLAY PROD1, TRADE1,PRODUCT2;

OPTION DECIMALS = 8;
DISPLAY OTHER1, PRICES;

* Parameters AS INDEX WITH 1993=1.000
PARAMETERS
* -- Price block
  IPL                 Wage rate index
  IPK(i)              Rent to  capital index
  IPT(ag)              Land rent index
  IPR(i)              Regional price index
  IP(i)               Composite price index
  IPG                 General composite price index
* -- Production block
  IL(i)               Labor demand index
  ITL                 Total labor demand index
  ILS                 Labor supply index
  IK(i)               capital demand index
  ITK                 Total Capital use index
  ITT                 Total Land use index
  IT(ag)               Land demand index
  IVA(i)              Value added index
  IX(i)               Output index
  ITVA              Total Value added index
  ITX               Total Output index
  ITE               Total Export index
  ITR               Total Reg. supply index
  ITM               Total Import index
  IVM(j,i)            Imported interm demand index
  IVR(j,i)            Regional interm demand index
  IR(i)               Regional supply index
  IE(i)               Export index
  IM(i)               Import index

* -- Income block

  IYH               Household (in the region) income index
  YHch              Change in hh income
  IDYH              Disposable income index
```

```
  IHSAV           Household saving index
  IYGOV             Government revenue index
  NETGOV            Net Revenue for government
  IGRP              Gross region product index
  GRPch             Change in Gross regional product
  CapComp           Capital Compensation
  LandComp          Land Compensation
  Rconsup           Resident angler consumer surplus loss
  NRconsup          NonResident angler consumer surplus loss
* -- Expenditure block
  IAHEXP          adj. Household expenditure index
  IGOVEXP         Government expenditure index
  IQ(i)           Commodity demand index
  IQM(i)          Imported commodity demand index
  IQR(i)          Regional commodity demand index
;

*-- EQUATIONS FOR CALCULATION OF INDEX WITH 1993=1.000

*### Price block
 IPL        = PL.L/PL0;
 IPK(i)     = PK.L(i)/PK0(i);
 IPT(ag) = PT.L(ag)/PT0(ag);
 IPR(i)     = PR.L(i)/PR0(i);
 IP(i)      = P.L(i)/P0(i);
 IPG                                                  =SUM(i,
(PR.L(i)*R0(i)+PM0(i)*M0(i))/(R0(i)+M0(i)))/4;
*#*  Production block
  IL(i)     = LAB.L(i)/L0(i);
  ITL       = (Sum(i,LAB.L(i))+(LHHH0+LGOV0))
                       /(Sum(i,L0(i))+LHHH0+LGOV0);
  ILS       = LS.L /LS0 ;
  IK(i)    = CAP.L(i)/K0(i);
  ITK       = Sum(i,PK.L(i)*CAP.L(i))/Sum(i,K0(i));
  IT("Agr") = LAND.L("Agr")/T0("Agr");
  ITT       = PT.L("Agr")*LAND.L("Agr")/T0("Agr");
  IVA(i)   = VA.L(i)/Va0(i);
  ITVA      = Sum(i,VA.L(i))/Sum(i,Va0(i));
  IX(i)     = X.L(i)/X0(i);
  ITX       =Sum(i,X.L(i))/Sum(i,X0(i));
  ITR       =Sum(i,R.L(i))/Sum(i,R0(i));
  ITM       =Sum(i,M.L(i))/Sum(i,M0(i));
  IVM(j,i)= VM.L(j,i)/VM0(j,i);
  IVR(j,i)= VR.L(j,i)/VR0(j,i);
  IR(i)    = R.L(i)/R0(i);
  IE(i)    = EXP.L(i)/E0(i);
  ITE        =Sum(i,EXP.L(i))/Sum(i,E0(i));
*## Income block
  IYH       = YH.L /YH0 ;
  IDYH      = DYH.L /DYH0 ;
  IHSAV     = HSAV.L /HSAV0 ;
  IGRP      = GRP.L/GRP0;
  GRPch     = GRP.L-GRP0;
*#Expenditure block
  IAHEXP    = AHEXP.L /HEXP0 ;
  IQ(i)    = Q.L(i)/Q0(i);
  IQM(i)   = QM.L(i)/QM0(i);
  IQR(i)   = QR.L(i)/QR0(i);
  IM(i)    = M.L(i)/M0(i);
  YHch      = YH.L -adjL.L*YH0 ;
  IYGOV     = YGOV.L/YGOV0;
  IGOVEXP   = GOVEXP.L/GOVEXP0;
  NETGOV    = YGOV.L-GOVEXP.L;
```

```
*##- SOLUTION VALUES OF INDEX
option decimals=5;

PARAMETER INDEX INDEXES FOR THE SIMULATION;
INDEX(I,"IPR")=IPR(I);
INDEX(I,"IX")=IX(I);
INDEX(I,"IE")=IE(I);
INDEX(I,"IL")=IL(I);
INDEX(I,"IK")=IK(I);
INDEX(I,"IPK")=IPK(I);
INDEX(ag,"IPT")=IPT(ag);
INDEX(ag,"IT")=IT(ag);
INDEX(I,"IVA")=IVA(I);
INDEX(I,"IR")=IR(I);
INDEX(I,"IM")=IM(I);
INDEX(I,"IQ")=IQ(I);
INDEX(I,"IQR")=IQR(I);
INDEX(I,"IQM")=IQM(I);
INDEX(I,"IPR")=IPR(I);
INDEX(I,"IPR")=IPR(I);

DISPLAY INDEX;
DISPLAY ITX,ITE,ITL,IPL,
         ITK,ITT,
         IGRP,GRPch,ITVA,ITR,ITM, YHch,
         IYH, IYGOV,IGOVEXP,NETGOV,
         ILS,IDYH,IHSAV,IAHEXP,
         IVM,IVR;

DISPLAY IGRP,IPG,IYH,ITE,ITM;
```