MPSGE:
Structure and Syntax

MPSGE: Generator of GE Models in MCP Syntax

Instead of using standard GAMS language an MCP can be formulated using MPSGE (a Mathematical Programming System for General Equilibrium) as an interface. Key advantages of MPSGE/GAMS include:

- Transparent and easy way to write down and analyze complicated systems of nonlinear inequalities: nonlinear equations in an MPSGE model are automatically generated from a tabular description of CES cost and expenditure functions --> MPSGE

- Use of vector syntax for large models ("classes" of functions) --> GAMS

- Convenient data handling (front-end) and report writing (back-end) --> GAMS

MPSGE - Key Elements

Declaration of parameters and values (GAMS)

$ONTEXT

Model declaration ($MODEL: <name>)

Declaration of sectors, commodities, consumers, auxiliary constraints

Production blocks

Demand blocks

Constraint equations

Report variable definition

$OFFTEXT

Include and solve statements

Other GAMS calculations (report)

MCP/GAMS - MPSGE: Key Relationships

Variables:

MCP:	MPSGE
prices	$commodities: a good or factor; the variable associated with a commodity is its price, not its quantity.
activities	$sectors: production activities that convert commodity inputs into commodity outputs; the variable associated with a sector is the activity level
consumers	$consumers:Individuals who supply factors and receive factor income, tax revenues, markups, and pay subsidies. In imperfectly competitive models, firm owners can be designated as consumers. The variable associated with a consumer is income from all sources.
"variables"	$auxiliary: Additional variables such as markup formulas or taxes with endogenous values which are functions of other variables such as prices and quantities.
report	$report: definition of report variables which can not be used as endogenous model variables within MPSGE (!).

GAMS/MCP - MPSGE: Key Relationships

MCP:	**MPSGE:**
zero-profit	$prod:
income definition	$demand:
market clearance	automatically generated
price/quantity constraints	$constraint:
reference price p	P:
reference quantity x	Q:
tax agent	A:
tax	T:
endogenous tax	M: (multiplier) N: (endogenous tax)
rationing multiplier	R:
elasticities	s: (substitution) t: (transformation)

MPSGE Calibration: P-field, Q-field, Elasticity

The elasticities together with the reference quantities and reference prices of inputs and outputs completely characterize the underlying nested CES functions (technologies or preferences).

MPSGE uses the information of a reference point to calibrate share form cost and expenditure functions:

(i) benchmark demand quantities (Q-fields)
--> provide an anchor point for isoquants / indifference curves

(ii) benchmark demand prices (P-fields)
--> fix the slope of the curve at that point

(iii) elasticity of substitution (exogenous) (s-fields)
--> describes the curvature of the indifference curve (second order approximation)

Note: No other data field in the $PROD: or $DEMAND: block alters the technology or preferences. E.g.: A tax rate change in a counter-factual experiment has no effect on the reference price.

$TITLE: CLOSED ECONOMY 2*2 IN MPSGE SCALAR SYNTAX

File: MPS_BASE.GMS

```
*       Formulation in MPSGE SCALAR SYNTAX
*       (NO USE OF GAMS FRONT-END FOR DATA HANDLING)

$ONTEXT
$MODEL:BASIC

$SECTORS:
        X    ! Activity level for X sector
        Y    ! Activity level for Y sector
        W    ! Activity level for utility production(Hicksian welfare
index)


$COMMODITIES:
        PX   ! Price index for X commodity
        PY   ! Price index for Y commodity
        PL   ! Price index for labor
        PK   ! Price index for capital
        PW   ! Price index for welfare (expenditure funtion)

$CONSUMERS:
        RA   ! Consumer income level

$PROD:X s:1
        O:PX    Q:100
        I:PL    Q:40
        I:PK    Q:60

$PROD:Y s:1
        O:PY    Q:100
        I:PL    Q:60
        I:PK    Q:40

$PROD:W s:1
        O:PW    Q:200
        I:PX    Q:100
        I:PY    Q:100

$DEMAND:RA
        D:PW    Q:200
        E:PL    Q:100
        E:PK    Q:100

$OFFTEXT

$SYSINCLUDE mpsgeset BASIC

        BASIC.WORKSPACE = 4;

* Do benchmark replication:
        BASIC.ITERLIM = 0;
$INCLUDE BASIC.GEN
        SOLVE BASIC USING MCP;
```

```
$TITLE: CLOSED ECONOMY 2*2 IN MPSGE VECTOR SYNTAX
```

File:mps_basv.gms

```
$ontext
The stylized economy is described as follows:
(i)     Agents: two producers, one representative household
(ii)    Behaviorial rules:
        producers maximize profits subject to technological constraints
        consumers maximize utility subject to budget constraint
(iii)   Market structure: perfectly competitive markets
(iv)    Technologies, preferences and resource endowments
        Two factors: capital K and labour L
        CD functions for producer (goods) S2 and S2 with inputs K & L
        Cobb-Douglas utility function in goods S1 and S2
(v)     Benchmark equilibrium as given below
$offtext


*SAM: ROWS ARE COMMDITIES, COLUMNS ARE ACTIVITIES AND DEMANDS

TABLE BMFLOW(*,*) BENCHMARK FLOWS

               X       Y        W       CONS     ROWSUM
  X          100             -100                    0
  Y                  100     -100                    0
  UTIL                        200      -200          0
  L          -40     -60                100          0
  K          -60     -40                100          0
* COLSUM       0       0        0         0
                                                       ;


*       Formulation in MPSGE vector syntax
*       Use of GAMS as front-end for data handling

SETS    I       Produced goods  /X, Y/,
        F       Factors pr production /L labour, K capital/;

ALIAS (FF,F), (I,J);


PARAMETERS      YBAR(I)         Benchmark output levels
                FDBAR(F,I)      Benchmark factor inputs
                CBAR(I)         Benchmark final demands
                ENDOW(F)        Factor endowments
                UBAR            Benchmark expenditure (consumption);

*     Extract values from the table:
YBAR(I)         = BMFLOW(I,I);
FDBAR(F,I)      = -BMFLOW(F,I);
CBAR(I)         = -BMFLOW(I,"W");
UBAR            = SUM(I,CBAR(I));
ENDOW(F)        = BMFLOW(F,"CONS");
```

$TITLE: CLOSED ECONOMY 2*2 IN MPSGE VECTOR SYNTAX (continued)

File:mps_basv.gms

```
*---------- Formulation in MPSGE ---------------------------------------------

$ONTEXT
$MODEL:BASIC

$SECTORS:
        Z(I)   ! Activity level for sectors
        U      ! Activity level for utility production(Hicksian welfare index)

$COMMODITIES:
        P(I)    ! Price index for commodities
        PF(F)   ! Price index for factors
        PW      ! Price index for welfare (expenditure funtion)

$CONSUMERS:
        RA      ! Representative consumer

$PROD:Z(I) s:1
        O:P(I)        Q:YBAR(I)
        I:PF(F)       Q:FDBAR(F,I)

$PROD:U  s:1
        O:PW          Q:UBAR
        I:P(I)        Q:CBAR(I)

$DEMAND:RA
        D:PW          Q:UBAR
        E:PF(F)       Q:ENDOW(F)

$OFFTEXT

$SYSINCLUDE mpsgeset BASIC

        BASIC.WORKSPACE = 4;

* Do benchmark replication check
        BASIC.ITERLIM = 0;
$INCLUDE BASIC.GEN
        SOLVE BASIC USING MCP;
```

MPSGE Calibration: Taxes and Reference Prices

Benchmark taxes are considered in the calibration of the P-field.

For a proper calibration, the value in the P: field depends on the benchmark value of the T: field (i.e.: benchmark taxes are reflected in the benchmark prices). Remember, however, that subsequent tax rate changes (in a counter-factual) do not change the underlying technology and thus leave the P: field unaltered.

Tax rates are interpreted differently for inputs and outputs.

- tax rates on inputs are specified on a <u>net</u> basis
  The user cost *uc* (reference price which goes into the P-field) of an *input* with market price *p* subject to an ad-valorem tax at rate *t* is:

  $uc = p\ (1 + t)$

- tax rates on outputs are specified on a <u>gross</u> basis.:
  The user cost *uc* (reference price which goes into the P-field) of an *output* subject to an ad-valorem tax at rate *t* is:

  $uc = p\ (1 - t)$

Logic: A tax *increases* the producer cost of inputs and *decreases* the producer value of outputs.

Example:

```
        TABLE BMFLOW(*,*) BENCHMARK FLOWS
                    X         Y          CONS
  X               100                      -100
  Y                         100            -100
  L               -20       -60              80
  K               -60       -40             100
  L_TAX           -20                        20 ;

$PROD:X s:1
        O:PX          Q:XBAR
        I:PL          Q:LBAR_X    P:2    A:CONS    T:1
        I:PK          Q:KBAR_X
```

```
$TITLE: CLOSED ECONOMY 2*2 WITH BENCHMARK TAXES IN MPSGE
```

File: mps_tax.gms

```
$ontext
This example illustrates the treatment of BMK taxes on inputs and
outputs. It is important to be aware that tax rates are interpreted
differently for inputs and outputs:

- tax rates on inputs are specified on a net basis
uc = p*(1 + t)

- tax rates on outputs are specified on a gross basis
uc = p*(1 - t)


MPSGE uses references prices and quantities for function calibration.
Taxes do not change the underlying technologies or preferences. The important
lesson is to keep track of what reference prices (and related quantities)
firms
and consumer face in the benchmark.
$offtext



TABLE BMFLOW(*,*) BENCHMARK FLOWS
               X       Y        W        CONS     ROWSUM
  X          100            -100                       0
  Y                  100    -100                       0
  UTIL                       200       -200            0
  L          -10     -60                 70            0
  K          -60     -40                100            0
  TAX_L      -10                         10
  TAX_PROD   -20                         20
* COLSUM       0       0        0         0 ;
```

$TITLE: CLOSED ECONOMY 2*2 WITH BENCHMARK TAXES IN MPSGE (continued)

File: mps_tax.gms

```
*---------- Formulation in MPSGE - Scalar Syntax ----------------------------

$ONTEXT
$MODEL:BMK_TAX_S

$SECTORS:
        X       ! Activity level for sector X
        Y       ! Activity level for sector Y
        W       ! Activity level for sector W (Hicksian welfare index)

$COMMODITIES:
        PX      ! Price index for commodity X
        PY      ! Price index for commodity Y
        PL      ! Price index for primary factor L
        PK      ! Price index for primary factor K
        PW      ! Price index for welfare (expenditure function)

$CONSUMERS:
        CONS    ! Income level for consumer CONS

$PROD:X s:1
        O:PX  Q:100     P:0.8   A:CONS  T:0.2
        I:PL  Q:10      P:2     A:CONS  T:1
        I:PK  Q:60

$PROD:Y s:1
        O:PY  Q:100
        I:PL  Q:60
        I:PK  Q:40

$PROD:W s:1
        O:PW   Q:200
        I:PX   Q:100
        I:PY   Q:100

*       Here we represent final excess demand.  This function
*       represents preferences (using reference demands), and
*       initial factor endowments:

$DEMAND:CONS
        D:PW   Q:200
        E:PL   Q: 70
        E:PK   Q:100

$OFFTEXT

$SYSINCLUDE mpsgeset BMK_TAX_S
        BMK_TAX_S.WORKSPACE = 4;

* Do a benchmark replication check:
        BMK_TAX_S.ITERLIM = 0;
```

$TITLE: CLOSED ECONOMY 2*2 WITH BENCHMARK TAXES IN MPSGE (continued)
File: mps_tax.gms

```
*---------- Formulation in MPSGE - Vector Syntax ----------------------------
--
$ontext
It might be usefuld to write (large-scale) vector syntax models it is helpful
to write models from "inside-out":
(i)     First, write down the MPSGE model
(ii)    Second, define associated SETS, PARAMETERS and SCALARS outside in the
        GAMS environment
$offtext


SETS    I       Produced goods  /X, Y/,
        F       Factors of production /L labour, K capital/;

ALIAS (FF,F), (I,J);


PARAMETERS      YBAR(I)         Benchmark output levels
                FDBAR(F,I)      Benchmark factor inputs
                CBAR(I)         Benchmark final demands
                ENDOW(F)        Factor endowments
                UBAR            Benchmark expenditure (consumption)
                TAX_IN(F,I)     Input tax on factors
                TAX_OUT(I)      Output tax on commodities
                PF_BAR(F,I)     Reference factor price
                PI_BAR(I)       Reference output price;

*     Extract values from the table:
YBAR(I)         = BMFLOW(I,I);
FDBAR(F,I)      = -BMFLOW(F,I);
CBAR(I)         = -BMFLOW(I,"W");
UBAR            = SUM(I,CBAR(I));
ENDOW(F)        = BMFLOW(F,"CONS");

TAX_IN(F,I)     = 0;
TAX_OUT(I)      = 0;
PF_BAR(F,I)     = 1;
PI_BAR(I)       = 1;


TAX_OUT(I)      = - BMFLOW("TAX_PROD",I)/BMFLOW(I,I);
PI_BAR(I)       = 1 - TAX_OUT(I);


TAX_IN("L",I)   = - BMFLOW("TAX_L",I)/(-BMFLOW("L",I));
PF_BAR(F,I)     = 1 + TAX_IN(F,I);

DISPLAY PF_BAR, PI_BAR, TAX_IN, TAX_OUT;
```

$TITLE: CLOSED ECONOMY 2*2 WITH BENCHMARK TAXES IN MPSGE (continued)
File: mps_tax.gms

```
*---------- Formulation in MPSGE - Vector Syntax ----------------------------

$ONTEXT
$MODEL:BMK_TAX_V

$SECTORS:
        Z(I)   ! Activity level for sectors
        U      ! Activity level for utility production(Hicksian welfare index)

$COMMODITIES:
        P(I)  ! Price index for commodities
        PF(F) ! Price index for factors
        PU    ! Price index for welfare (expenditure funtion)

$CONSUMERS:
        RA      ! Representative consumer

$PROD:Z(I) s:1
        O:P(I)        Q:YBAR(I)    P:PI_BAR(I)   A:RA T:TAX_OUT(I)
        I:PF(F)       Q:FDBAR(F,I) P:PF_BAR(F,I) A:RA T:TAX_IN(F,I)

$PROD:U s:1
        O:PU         Q:UBAR
        I:P(I)       Q:CBAR(I)

$DEMAND:RA
        D:PU           Q:UBAR
        E:PF(F)        Q:ENDOW(F)

$OFFTEXT

$SYSINCLUDE mpsgeset BMK_TAX_V

        BMK_TAX_V.WORKSPACE = 4;

* Do the benchmark replication
        BMK_TAX_V.ITERLIM = 0;
$INCLUDE BMK_TAX_V.GEN
        SOLVE BMK_TAX_V USING MCP;


* Set taxes in the counterfactual to zero
TAX_OUT(I)  = 0;
TAX_IN(F,I) = 0;


        BMK_TAX_V.ITERLIM = 2000;
$INCLUDE BMK_TAX_V.GEN
        SOLVE BMK_TAX_V USING MCP;
```

Example: Nested CES Cost Functions

--> sources: mcp_nest.gms, mps_nest.gms

This example illustrates how convenient MPSGE is for the formulation of nested CES functional forms as compared to the standard MCP syntax. Whereas in the MCP form we have to write down tedious cost and demand functions, these formulae are automatically generated within MPSGE.

Key rules:

The nesting structure is indicated on the first line of the $PROD block (or $DEMAND block).

For difficult nesting structures, it is possible to make references between different substitution fields.

As to syntax issues regarding aggregate labels: Names *s:* and *t:* are reserved for top level input and output (substitution and transformation) aggregates.

```
$PROD:X s:0.5  KY:1
        O:PX  Q:120
        I:PY    Q:20  KY:
        I:PL    Q:40  A:CONS  T:TL_X
        I:PK    Q:60  KY:

$PROD:Y s:0.75  KX:1
        O:PY     Q:120
        I:PX     Q: 20  KX:
        I:PL     Q: 60  A:CONS  T:TL_Y
        I:PK     Q: 40  KX:
```

Task: Write down the MCP cost functions for sector x and sector y and the associated factor demands for sectoral production!

$TITLE: CLOSED ECONOMY 2*2 WITH SEPARABLE NESTED CES-FUNCTIONS IN MCP SYNTAX

File: mcp_nest.gms

```
$ontext
This example illustrates that the MCP formulation of GE problems can
get rather tedious when nested functional forms are employed (need to apply
chain rule for the derivation of compensated demand functions).


Here we assume 2 level separable nested CES functions in production.

*       The X sector production function is a two-level CES.
*       In the KY, K and Y form a Cobb-Douglas aggregate.
*       In the top level, L and KY(K,Y) have an elasticity of
*       substitution equal to 0.5

*       The Y sector production function is a two-level CES.
*       In the KX, K and X form a Cobb-Douglas aggregate.
*       In the top level, L and KX(K,X) have an elasticity of
*       substitution equal to 0.5


For the specification of NNCES functions which allow the use of
arbitrary AUES between inputs see Rutherford/Perroni (EER 1997).

                 Production Sectors          Consumers
   Markets  |     X       Y        W    |       CONS
   ------------------------------------------------------
       PX   |  120     -20     -100    |
       PY   |  -20     120     -100    |
       PW   |                   200    |        -200
       PL   |  -40     -60             |         100
       PK   |  -60     -40             |         100
    ------------------------------------------------------
$offtext
```

$TITLE: CLOSED ECONOMY 2*2 WITH SEPARABLE NESTED CES-FUNCTIONS IN MCP SYNTAX (continued)

File: mcp_nest.gms

```
PRF_X.. 120 * ( 1/3 * (PL*(1+TL_X))**(1-0.5) +
                2/3 * (PK**0.75 * PY**0.25)**(1-0.5) )**(1/(1-0.5))
                        =E= 120 * PX;

PRF_Y.. 120 * ( 1/2 * (PL*(1+TL_Y))**(1-0.75) +
                1/2 * (PK**(2/3)* PX**(1/3))**(1-0.75) )**(1/(1-0.75))
                        =E= 120 * PY;

PRF_W.. 200 * PX**0.5 * PY**0.5 =E= 200 * PW;

MKT_X.. 120 * X =E= 100 * W * PX**0.5 * PY**0.5 / PX +
     20*Y*(PY/(PK**(2/3)* PX**(1/3)))**0.75* PK**(2/3)* PX**(1/3)/PX;

MKT_Y.. 120 * Y =E= 100 * W * PX**0.5 * PY**0.5 / PY +
     20*X*(PX/(PK**0.75* PY**0.25))**0.5* PK**0.75* PY**0.25/PY;

MKT_W.. 200 * W =E= RA / PW;

MKT_L.. 100  =E= 40 * X * (PX/((1+TL_X)*PL))**0.5 +
                 60 * Y * (PY/((1+TL_Y)*PL))**0.75;

MKT_K.. 100 =E=  60 * X * (PX/(PK**0.75 * PY**0.25))**0.5
                       * (PK**0.75 * PY**0.25) /PK
                 + 40 * Y *(PY/(PK**(2/3) * PX**(1/3)))**0.75
                       * PK**(2/3) * PX**(1/3) /PK;

I_RA.. RA =E= 100*PL + 100*PK +
  TL_X * PL *40 * X * (PX/((1+TL_X)*PL))**0.5 +
  TL_Y * PL *60 * Y * (PY/((1+TL_Y)*PL))**0.75;

MODEL ALGEBRAIC /PRF_X.X, PRF_Y.Y, PRF_W.W, MKT_X.PX, MKT_Y.PY, MKT_L.PL,
                 MKT_K.PK, MKT_W.PW, I_RA.RA /;

*       Check the benchmark:

        X.L=1; Y.L=1; W.L=1; PL.L = 1;PX.L=1; PY.L=1; PK.L=1; PW.L=1;
RA.L=200;

        TL_X = 0; TL_Y = 0;
        SOLVE ALGEBRAIC USING MCP;

*       Solve the same counterfactual:

        TL_X = 1; TL_Y = 0.5;
        SOLVE ALGEBRAIC USING MCP;
```

Example: Multi-level nested CES Functions

Example: Nested structure to account for different substitution possibilites across fuels in various sectors. A sophisticated structure may be important to reflect the "real-world" scope for interfuel substitution. These substitution patterns are a decisive determinant of the costs of environmental regulation with respect to energy-related emissions

```
SETS
          S          Goods     /AAA*AAZ, HCO, SCO, ELE, GAS, OIL/,
          EG(S)      Energy goods /HCO, SCO, ELE, GAS, OIL/,
          COAL(S)    Hard and soft coal inputs into energy aggregate
                     /HCO, SCO/,
          ELEC(S)    Electricity input into energy aggregate
                     /ELE/,
          NELE(S)    Gas & oil inputs into energy aggregate
                     /GAS, OIL/;

PARAMETERS
          ESUB_NUCL(S)  ELASTICITY OF NUCLEAR VERSUS COAL_ELE_OIL_GAS,
          ESUB_COAL(S)  ELASTICITY OF HARD&SOFT COAL VERSUS ELE_OIL_GAS,
          ESUB_ELEC(S)  ELASTICITY OF ELECTRICITY VERSUS OIL_GAS,
          ESUB_NELE(S)  ELASTICITY OF OIL VERSUS GAS;

$PROD:E(S)              s:ESUB_NUCL(S)
+         a:ESUB_COAL(S) b(a):ESUB_ELEC(S) c(b):ESUB_NELE(S)
      O:PE(S)           Q:E0(S)
      I:PEG(EG,S)       Q:D0E(EG,S)   a:$COAL  b:$ELEC  c:$NELE
      I:PNUC            Q:D0NUC(S)
```

For the specification of non-separable nested CES functions, which allow the use of arbitrary Allen-Uzawa elasticities of substitution between inputs, see Rutherford and Perroni 1997 (CES demand functions).