MPSGE Version 2

Thomas F. Rutherford

Wisconsin Institute for Discovery
Department of Agricultural and Applied Economics
University of Wisconsin, Madison

WiNDC Advisory Board Meeting
Madison, Wisconsin

23 April, 2021

Collaborators

- James Markusen (University of Colorado Boulder)
- Florian Landis (ZEW Mannheim)

MPSGE: A Mathematical Programming System for General Equilibrium Analysis

- Model representation tool for a specific class of economic models: Arrow-Debreu general equilibrium.
- Incorporates facilities for automatic calibration of cost and expenditure functions
- Model input is *tabular* (non-algebraic) and follows the broad schematic structure of the MPS format for general equilibrium models
- Provides routines for providing demand and supply functions (price-responsive netputs for production sectors and excess demands for consumers) and *analytic Jacobians*.
- Provides background error checks related to model consistency

Orchard Hayes' MPS Format for Linear Programming

```
NAME          TESTPROB
ROWS
 N  COST
 L  LIM1
 G  LIM2
 E  MYEQN
COLUMNS
    XONE      COST                 1   LIM1                 1
    XONE      LIM2                 1
    YTWO      COST                 4   LIM1                 1
    YTWO      MYEQN               -1
    ZTHREE    COST                 9   LIM2                 1
    ZTHREE    MYEQN                1
RHS
    RHS1      LIM1                 5   LIM2                10
    RHS1      MYEQN                7
BOUNDS
 UP BND1      XONE                 4
 LO BND1      YTWO                -1
 UP BND1      YTWO                 1
ENDATA
```

The MPSGE Format for General Equilibrium

```
$MODEL:M2X2

$SECTORS:
X       ! Capital-intensive activity level
Y       ! Labor-intensive activity level

$COMMODITIES:
PX      ! Price of capital-intensive goods
PY      ! Price of labor-intensive goods
PW      ! Wage rate
PS      ! Return to capital (skills)

$CONSUMERS:
CONS    ! Representative agent

$PROD:X S: 1
O:PX    Q: 100  A:CONS  T:-0.1
I:PW    Q: 40
I:PS    Q: 60

$PROD:Y S: 1
O:PY    Q: 100
I:PW    Q: 60
I:PS    Q: 40

$DEMAND:CONS    S: 1
D:PX    Q: 100
D:PY    Q: 100
E:PW    Q: 100
E:PS    Q: 100

$SOLVE
```

MPSGE Historical Timeline

1981 Lars Mathiesen spends sabbatical at the Stanford OR Department with the research objective of implementing Newtons method for generalized equations [Josephy (1979)]

1982 A pilot implementation of the nonlinear complementarity solver MILES is completed, based in Tomlin's LCPL code for Lemke's algorithm and Saunders' LUSOL (the sparse matrix factorization code from MINOS)

1982 A trade policy research project, *Market Prospects*, is undertaken at NHH in Bergen. A central element of the project is a global Heckscher-Ohlin trade model, VEMOD. Project participants included Victor Norman, Agnar Sandmo, Lars Mathiesen, Terje Hansen, Terje Lensburg, and Erling Stigum.

1983 A standardized set of routines for representing nested CES functions is implemented to help with the ongoing formulation and reformulation of VEMOD.

1984 A pilot implementation of MPSGE is presented at TIMS XXVI, June 17-21, 1984.

The Impetus for GAMS (Alex Meeraus)

"GAMS's impetus for development arose from the frustrating experience of a large economic modeling group at the World Bank. In hindsight, one may call it a historic accident that in the 1970s mathematical economists and statisticians were assembled to address problems of development. They used the best techniques available at that time to solve multi sectoral economy-wide models and large simulation and optimization models in agriculture, steel, fertilizer, power, water use, and other sectors. Although the group produced impressive research, initial success was difficult to reproduce outside their well functioning research environment. The existing techniques to construct, manipulate, and solve such models required several manual, time-consuming, and error-prone translations into different, problem-specific representations required by each solution method."

GAMS Historical Timeline

1976 GAMS idea is presented at the ISMP Budapest

1978 Phase I: GAMS supports linear programming. Supported platforms: Mainframes and Unix Workstations

1979 Phase II: GAMS supports nonlinear programming.

1987 GAMS becomes a commercial product

1988 First PC System (16 bit)

1989 GAMS begins to be used as a front-end and back-end to MPSGE, producing input data matrices and model reports.

1991 Alex Meeraus collaborates on implementation of MPSGE and MCP as GAMS subsystems

1994 GAMS supports mixed complementarity problems

The GAMS/MPSGE Format

```
$ontext
$model:gtap8mge

$sectors:
        y(g,r)$vom(g,r)                  ! Supply
        m(i,r)$vim(i,r)                  ! Imports
        ft(f,r)$(sf(f) and evom(f,r))    ! Specific factor transformation

$commodities:
        p(g,r)$vom(g,r)                          ! Domestic output price
        pm(j,r)$vim(j,r)                         ! Import price
        pf(f,r)$evom(f,r)                        ! Primary factors rent
        ps(f,g,r)$(sf(f) and vfm(f,g,r))         ! Sector-specific primary factors

$consumers:
        ra(r)                   ! Representative agent

$prod:y(g,r)$vom(g,r)   s:esub(g)    i.tl:esubd(i)  va:esubva(g)
        o:p(g,r)        q:vom(g,r)      a:ra(r)  t:rto(g,r)
        i:p(i,r)        q:vdfm(i,g,r)   p:(1+rtfd0(i,g,r)) i.tl:  a:ra(r) t:rtfd(i,g,r)
        i:pm(i,r)       q:vifm(i,g,r)   p:(1+rtfi0(i,g,r)) i.tl:  a:ra(r) t:rtfi(i,g,r)
        i:ps(sf,g,r)    q:vfm(sf,g,r)   p:(1+rtf0(sf,g,r))  va:    a:ra(r) t:rtf(sf,g,r)
        i:pf(mf,r)      q:vfm(mf,g,r)   p:(1+rtf0(mf,g,r))  va:    a:ra(r) t:rtf(mf,g,r)
...
$demand:ra(r)
        d:p("c",r)      q:vom("c",r)
        e:p("c",rnum)   q:vb(r)
        e:p("g",r)      q:(-vom("g",r))
        e:p("i",r)      q:(-vom("i",r))
        e:pf(f,r)       q:evom(f,r)

$offtext
$sysinclude mpsgeset gtap8mge
```

GAMS/MPSGE Automated Model Generation

```
$OFFLISTING

$OFFINLINE
$INLINECOM { }
PUT   MPS,'$MODEL:GTAP8MGE'/;

{    2}  PUT /;

{    2}  PUT'$SECTORS:'/;
{    3}  LOOP((G,R)$(VOM(G,R)),
{    3}    PUT /,'Y''.'G.TL'.'R.TL;
...

{   18}  LOOP((G,R)$(VOM(G,R)),
{   18}    PUT /;
{   18}    PUT /,'$PROD:Y''.'G.TL'.'R.TL;
{   18}    IF ((ABS(ESUB(G)) GT MPSEPS),PUT /,'+S:'ESUB(G);  );
{   18}    LOOP((I),
{   18}      PUT /,'+',I.TL,':'ESUBD(I);
{   18}      PUT /,'+VA:'ESUBVA(G);
{   19}    );
{   19}    IF(ABS(VOM(G,R)) GT MPSEPS,
{   19}      PUT /,'O:P''.'G.TL'.'R.TL;
{   19}      PUT /,'+Q:'VOM(G,R);
{   19}      PUT /'+','A:RA''.'R.TL;
{   19}      IF ((ABS(RTO(G,R)) GT MPSEPS),PUT /,'+T:'RTO(G,R);  );
{   20}    );
{   20}    LOOP((I),
{   20}      IF(ABS(VDFM(I,G,R)) GT MPSEPS,
{   20}        PUT /,'I:P''.'I.TL'.'R.TL;
{   20}        PUT /,'+Q:'VDFM(I,G,R);
...
```

GAMS/MPSGE Model Data: `GASMSCGE.DAT`

```
$PROD:Y.dwe.CHN dwe: 1.89999998E+00 oil: 2.09999990E+00 gas: 1.19062693E+01
+       omn: 8.99999976E-01 lum: 3.40000010E+00 ppp: 2.95000005E+00
...
 O:P.dwe.CHN Q: 1.19713412E+02 A:RA.CHN T: 3.29997896E-06
 I:P.dwe.CHN Q: 7.43021990E-01 dwe:
 I:P.oil.CHN Q: 4.78151689E-03 oil:
 I:P.omn.CHN Q: 7.32515085E-03 omn:
 I:P.lum.CHN Q: 1.01078711E-01 lum:
 I:P.ppp.CHN Q: 2.70655032E-01 ppp:
 I:P.crp.CHN Q: 1.17877632E+00 crp:
  ...
 I:PM.oil.CHN Q: 4.70509787E-04 oil:
 I:PM.omn.CHN Q: 1.72209354E-04 omn:
 I:PM.lum.CHN Q: 3.55582434E-03 lum:
 I:PM.ppp.CHN Q: 3.18539927E-02 ppp:
 I:PM.crp.CHN Q: 2.72512885E-01 crp:
  ...
 I:PF.lab.CHN Q: 1.57947255E+01 P: 1.00018299E+00 VA: A:RA.CHN T: 1.82986549E-04
 I:PF.cap.CHN Q: 8.38820394E+01 P: 1.00143996E+00 VA: A:RA.CHN T: 1.43995925E-03
```

GAMS/MPSGE: Code for Function Evaluations

```
      SUBROUTINE gempsa (gdx, gdbl, gdbu, ncol)

C     VALIDATE MODEL CONSISTENCY, COUNT THE NUMBER OF NONZEROS IN A
C     AND COMPLETE MEMORY ALLOCATION

      USE mgeParm
      USE mgeMem
      INCLUDE 'mpscom.inc'
      INTEGER, INTENT(IN) :: ncol
      REAL(KIND=8), DIMENSION(ncol) :: gdx, gdbl, gdbu

      CHARACTER(LEN=128) :: buf

C     CHECK CONSISTENCY OF MODEL STRUCTURE AND MOVE
C     REPORT VARIABLE POINTERS INTO THE FUNCTION WORKSPACE:
      CALL GECHKM(vSOURCE, vSINK, vFUNLOC,
     *     vVTYPE, NCOL, nFnDim, vFUNVEC, vFUNVEC)

C     SET WORKSPACE DIMENSIONS TO MATCH PROBLEM SIZE AND
C     FREE UP STORAGE FOR SOLUTION ALGORITHM:
      nFnDim = lStore

C     ALLOCATE CORE FOR LA, LN, IA AND JA:
      CALL memSetSizes2 (nFnDim, nRV, iaDim)
      IF (.NOT. memAlloc(2,buf)) THEN
         CALL GEERRR('Error allocating memory:' // trim(buf))
      END IF
```

GAMS/MPSGE: Code for Consistency Checks

```
C
C        CHECK FOR RATIONED ENDOWMENTS:
C
         IF (VTYPE(K).EQ.3) THEN
            DO 125 I=1,NE
               IG = IE(I)
               IF (E(I).GT.ZERO) SOURCE(IG) = .TRUE.
               IF (E(I).LT.ZERO) SINK(IG)   = .TRUE.
 125        CONTINUE
         ENDIF
C
C        FLAG HOUSEHOLDS WITHOUT FINAL DEMAND ITEMS.
C
         IF (VTYPE(K).EQ.3 .AND. NDEM.EQ.0) THEN
            CALL GFNAME(K,NB,NAME)
            WRITE(msgBuf,130) NAME (1:NB)
            CALL msgWrapper (msgLst, msgBuf)
 130        FORMAT(' **** Error: No consumption good for ',A)
            WARNING = .TRUE.
         ENDIF
```

Mathiesen's Equilibrium Format

Three sets of "central variables":

p a non-negative n-vector of commodity prices including all final goods, intermediate goods and primary factors of production, corresponding to MPSGE `$commodities`.,

y a non-negative m-vector of activity levels for constant returns to scale production sectors in the economy, corresponding to MPSGE `$sectors`,

M an h-vector of income levels, one for each "household" in the model, including any government entities, corresponding to MPSGE `$consumers`,

μ a k-vector of auxiliary variables which may determine consumer endowment vectors or tax rates, corresponding to MPSGE `$auxiliary`,

Equilibrium Conditions

- Zero Profit

$$-\Pi_j(p) = C_j(\tilde{p}) - R_j(\tilde{p}) \geq 0 \quad \perp y_j$$

where:

$$C_j(\tilde{p}) = \min \sum_j p_i(1 + t_{ij})x_i \quad \text{s.t.} \quad f_j(x) = 1$$

$$R_j(\tilde{p}) = \max \sum_j p_i(1 - \tau_{ij})y_i \quad \text{s.t.} \quad g_j(x) = 1$$

- Market Clearance

$$\sum_j y_j \frac{\partial \Pi_j(\tilde{p})}{\partial \tilde{p}_i} + \sum_h \omega_{ih} \geq \sum_h d_{ih}(p, M_h) \quad \perp p_i$$

where

$$d_{ih} = \text{argmax}_x U_h(x) \quad \text{s.t.} \quad \sum_i p_i x_i \leq M_h$$

Equilibrium Conditions (cont.)

Income arises both from primary factor endowment and from tax revenue:

- Income Balance

$$M_h = \sum_i p_i \omega_{ih} + \sum_{ij} y_j \theta_{ijh} p_i \left(t_{ij} x_{ij} + \tau_{ij} y_{ij} \right)$$

- Auxiliary constraints associated with *auxiliary variables* μ which may in term endogenize endowments (ω) or taxes (t or τ):

$$F_k(y, p, M, \mu) \geq 0 \quad \perp \mu_k$$

Auxiliary Variables

One key motivation for the development of GAMS/MPSGE was to provide *extensibility*. This arises in MPSGE models through the use of *auxiliary variables*:

- Auxiliary variables in MPSGE may enter enter model either in a dual context (as an endogenous tax) or a primal context (as a rationing instrument for household endowments).
- Each auxiliary variable has an associated equation which is written in GAMS algebra.
- Under the hood auxiliary equations are processed by GAMS while the remaining model equations are provided by MPSGE.

Sources of Confusion for Students of MPSGE

MPSGE as a modeling tool:
"A day to learn, a lifetime to master."

Students typically run into trouble with the following issues:

- Calibration of CES Functions
- Nested functions
- Margins (a set of nests)
- Multiple taxes on a single transaction
- Absence of algebraic GAMS code

CES/CET Functions

- Textbook treatment of constant elasticity functions

$$U(x, y) = (x^\rho + y^\rho)^{1/\rho}$$

- Research monograph treatment of constant elasticity functions

$$U(x, y) = \phi\left(\alpha x^\rho + (1-\alpha) y^\rho\right)^{1/\rho}$$

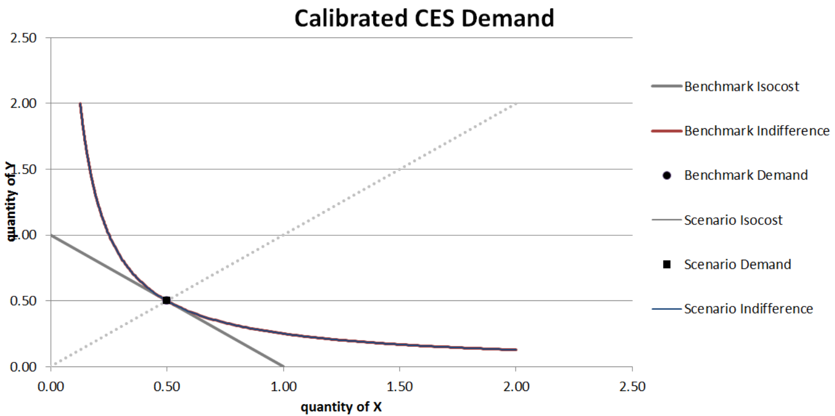
or

$$U(x, y) = (\alpha x^\rho + \beta y^\rho)^{1/\rho}$$

- Calibrated share form (money metric):

$$U(x, y) = \bar{M}\left(\theta_x \left(\frac{x}{\bar{x}}\right)^\rho + \theta_y \left(\frac{y}{\bar{y}}\right)^\rho\right)^{1/\rho}$$

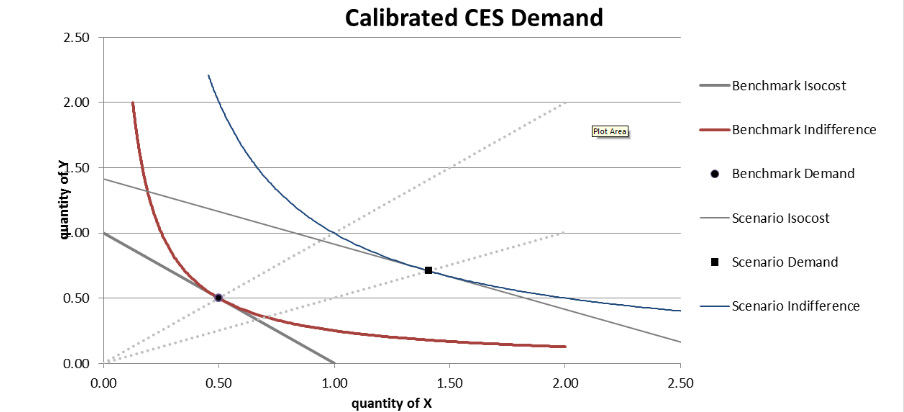
Calibration: Benchmark Equilibrium



x0 0.50	px0 1.0	sigma 1.0	epsilon 2	px 1	py 1
Benchmark Value Share of X	Benchmark Price of X (py0=1)	Elasticity of Substitution	Price Elasticity of Aggregate Demand	Scenario Price of X	Scenario Price of Y

Counterfactual Equilibrium

CESDemand - Microsoft Excel



x0 0.50	px0 1.0	sigma 1.0	epsilon 2	px 0.5	py 1
Benchmark Value Share of X	Benchmark Price of X (py0=1)	Elasticity of Substitution	Price Elasticity of Aggregate Demand	Scenario Price of X	Scenario Price of Y

Trade Activites in GTAP: Margins and Tax Agents

```
$prod:M(i,r)$vim(i,r)   s:esubm(i)  s.tl:0
        o:PM(i,r)       q:vim(i,r)

*       Imports from region s are assigned to nest s.tl:

        i:P(i,s)        q:vxmd(i,s,r)   p:pvxmd(i,s,r) s.tl:

*       Export subsidies accruing to region s:

+               a:RA(s) t:(-rtxs(i,s,r))

*       Import tariffs accruing to region r:

+               a:RA(r) t:(rtms(i,s,r)*(1-rtxs(i,s,r)))

*       Trade margin j for each import source s:

        i:PT(j)#(s) q:vtwr(j,i,s,r) p:pvtwr(i,s,r) s.tl:
+               a:ra(r) t:rtms(i,s,r)
```

Objectives for MPSGE Version 2

1. Generate model equations in self-documenting GAMS/MCP (algebraic) format.
2. Introduce *external effects* in MPSGE production functions.
3. Provide functionality for using MPSGE-generated demand and supply functions in auxiliary constraints or independent of MPSGE models.
4. Implement with programming tools amenable to open source approach (i.e., Not FORTRAN!)
5. Provide a simple, automated framework for "second-order senstivity analysis".
6. Explore alternative model formualations, e.g. using tools for *extended mathematical programming*.

MPSGE-V2 Model Equations

```
*   Definition of zero-profit conditions
mge_zprf_Y(g,r)$vom(g,r)..
     (mge_cbar_Y(g,r)*mge_CES_Y_s(g,r))$mge_cond_Y(g,r)
       =g=
         (mge_rbar_Y(g,r)*mge_CET_Y_t(g,r))$mge_cond_Y(g

mge_zprf_M(i,r)$vim(i,r)..
     (mge_cbar_M(i,r)*mge_CST(M,s,mge_elast_sinM(i,r),(i
       =g=
         (mge_rbar_M(i,r)*mge_CET_M_t(i,r))$mge_cond_M(i

mge_zprf_YT(j)$vtw(j)..
     (mge_cbar_YT(j)*mge_CD_YT_s(j))$mge_cond_YT(j)
       =g=
         (mge_rbar_YT(j)*mge_CET_YT_t(j))$mge_cond_YT(j)
```

External Effects – The X: Field

We want to provide for the use of auxiliary variables as productivity mulipliers, as in:

$$y = f(x) = \phi \left(\sum_i \alpha_i (\mu_i x_i)^\rho \right)^{1/\rho}$$

NB: scale and share parameters remain unchanged, and the productivity multiplier enhance the marginal product of the individual inputs. MPSGE syntax using the `x:` field:

```
$prod:Y  s:sigma
   o:PY    q:y0
   i:PX(i) q:x0(i)  p:p0(i)     x:MU(i)
```

Algebraic Details

- Calibrated share form:

$$y = f(x) = \bar{y} \left(\sum_i \theta_i \left(\frac{x_i \mu_i}{\bar{x}_i} \right)^{\rho} \right)^{1/\rho}$$

- Externality-adjusted unit cost:

$$C(p) = \bar{c} \left(\sum_i \theta_i \left(\frac{p_i}{\mu_i \bar{p}_i} \right)^{1-\sigma} \right)^{1/(1-\sigma)}$$

- Hicksian compensated demand functions:

$$x_i(p) = \frac{\bar{q}_i}{\mu_i} \left(\frac{C(p)}{\bar{c}} \frac{\mu_i \bar{p}_i}{p_i} \right)^{\sigma} y$$

Aside: The Jevons Paradox

Let us assume competitive markets for output with a price elasticity of demand equal to ϵ, i.e.:

$$y = \left(\frac{\bar{c}}{C(p)}\right)^{\epsilon}$$

In the present framework we can then assess the effect of a marginal improvement in the efficiency of an input x_i by evaluating the demand elasticity with respect to technical improvement::

$$\left.\frac{\partial x_i}{\partial \mu_i}\frac{\mu_i}{x_i}\right|_{\mu_i=1, x_j=\bar{x}_j \forall j} = \left.\frac{\partial x_i}{\partial \mu_i}\frac{1}{\bar{x}_i}\right|_{y=1} + x_i \frac{\partial y}{\partial C(p)}\frac{\partial C(p)}{\partial \mu_i} = \sigma - 1 + \epsilon\theta_i.$$

We see that in the single-level CES setting, the Jevons paradox is present when $\sigma + \theta_i \epsilon > 1$.

Demand and Supply Variables

```
$prod:m(i,r)$vim(i,r)   s:esubm(i)  s.tl:0

o:pm(i,r)    q:vim(i,r)
i:p(i,s)     q:vxmd(i,s,r) [QXMD(i,s,r)] p:pvxmd(i,s,r) s.tl:
+  a:ra(s) t:(-rtxs(i,s,r)) [VTXS(i,s,r)]
+  a:ra(r) [VTMS(i,s,r)]  t:(rtms(i,s,r)*(1-rtxs(i,s,r)))

i:pt(j)#(s) q:vtwr(j,i,s,r) [QTWR(j,i,s,r)] p:pvtwr(i,s,r) s.tl:
+  a:ra(r) t:rtms(i,s,r)
```

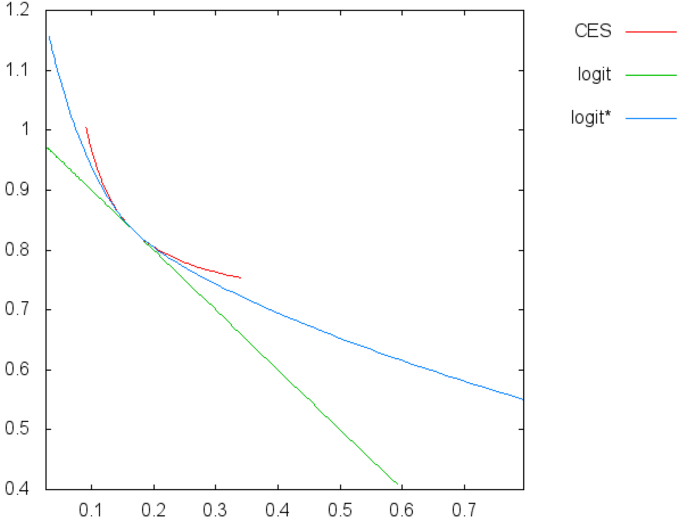
The Ruby Parser

```
  1,616   parse_methods.rb
  3,348   mpsge.gms
 61,020   mpsgeClasses.rb
 79,065   mpsgeparser.rb
  4,891   mpsgePatterns.rb

#########################################################################################
# GLOBAL LEVEL                                                                          #
#########################################################################################

Quiet=true
module Read_Methods
  require_relative "mpsgePatterns"
  include Patterns

  def readModel_scr(input,modelname)
    first_line=false
    last_line=false
    modelText=[]
    #LOOK FOR TEXT DESCRIBING DESIRED MODEL AND REMEMBER THE FIRST AND THE LAST LINE
    input.each_with_index do |line, i|
      if line =~ /^(\$MODEL:)\s*(#{modelname})\s*$/i
        first_line=i
...
module Patterns
  Number = /(?:[\+\-]?(?:\d+(?:\.\d*)?|\.\d+)(?:E[\+\-]?\d+)?|yes|no|[+-]?inf|na|eps)/i
  Comment = /\!.*/i
  Rcomment = /\!\s*(.*)/i
  Index = /(?:#{Name}|\'[^\']+\'|\"[^\"]+\")/i
  Rquotedindex = /^(\'[^\']+\'|\"[^\"]+\")$/i
  Rindices = /\s*\,?\s*(#{Index})((?:\s*\,\s*#{Index})*)/i
  Parvar = /#{Name}(?:\(\s*#{Index}(?:\s*\,\s*#{Index})*\))?/i
```

Second Order Sensitivity

- While flexilibe functional forms (Translog, Generalized Leontief, Normalized Quadratic etc.) could be calibrated to a benchmark equilibrium, extraction of the benchmark Slutsky matrix may be quite tedious.
- Three alternatives seem more easily programmed:
  - Johansen log-linearization (ala GEMPACK)
  - Equilibrium displacement (also logarithmic)
  - Nest logit demand and supply functions which can be precisely calibrated with the same input data used for the nested CES: benchmark value shares and nest-level elasticities of substitution and transformation.
- The object here is provide a framework for low-cost assessment of robustness wrt higher order elasticities (functional form).

Logit Versus CES: Indifference Curves



Logit Versus CES: Own Price Response