Introduction to GAMS: GAMSCHK

Dhazn Gillig
&
Bruce A. McCarl

Department of Agricultural Economics
Texas A&M University

What is GAMSCHK?

GAMSCHK is a system for verifying model structure and solutions to see if all is correct. It is designed to aid in PRE and POST solution model analysis and help fix improperly working models (see Chapter 17 in McCarl and Spreen text book for details).

Why do we need to use PRE solution with GAMS Check?

Answer: **To verify the model structure before worrying too much about the answer. GAMSCHK automatically checks a model for errors and portrays information about its structure in several ways before solving.**

Why do we need to use POST solution with GAMS Check?

Answer: **To enlist the solvers help in an exercise to find the causes of unrealistic solutions, or unbounded or infeasible problems.**

PRE Solution

PRE- solution procedures can be used to

(words in parentheses are GAMS check procedures)

- **List selected equations and/or variables (DISPLAYCR)**
- **Generate schematics on equations/variables blocks (BLOCKPIC)**
- **List characteristics of equations/variables blocks (BLOCKLIST)**
- **Find obvious specification errors (ANALYSIS)**
- **Generate schematics on location of coefficients by sign and magnitude on individual equation/variable basis (PICTURE)**
- **List characteristics of equations/variables (MATCHIT)**

POST Solution

POST- solution procedures are used to fix misbehaving models by.

(words in parentheses are GAMS check procedures)

- **Reconstructing reduced cost and equation activity (POSTOPT)**
- **Helping resolve problems with unbounded or infeasible models (NONOPT)**

On a post solution basis, POSTOPT is used to check for Non-Sensical solutions by observing a faulty attribute of the solution in terms of

- **Allocation (variable and equation levels, e.g. Variable.L, Equation.L)**
- **Valuation (variable and equation marginals e.g. Variable.M, Equation.M)**

Steps to run GAMS CHECK

Here are steps to run GAMSCHK.

Step 1: Insert a command line

OPTION LP = GAMSCHK ; for LP problem

or OPTION NLP = GAMSCHK ; for NLP problem

or OPTION MIP = GAMSCHK ; for MIP problem

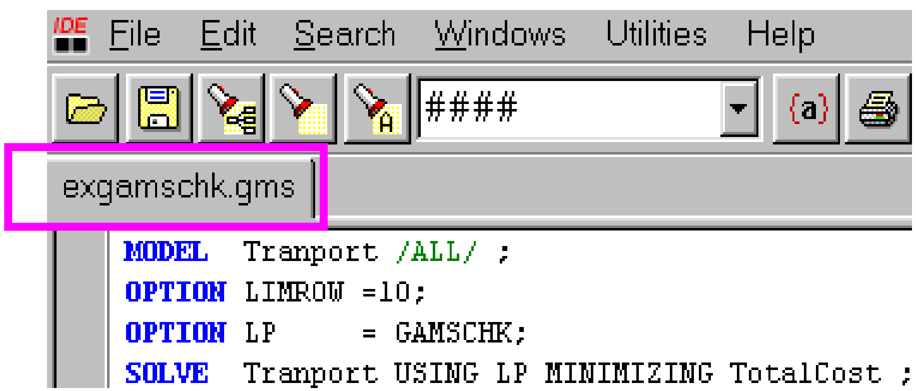
in the model right before the solve

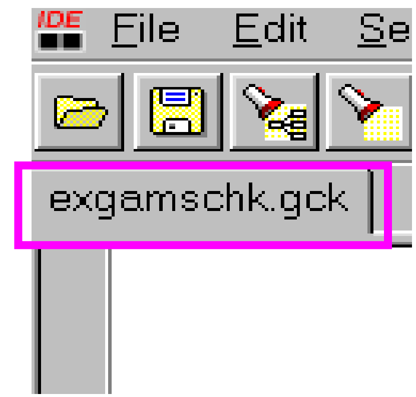
```
Model  Tranport /ALL/ ;
OPTION LP      = GAMSCHK;
Solve  Tranport USING LP MINIMIZING TotalCost ;
```

Steps to run GAMSCHK

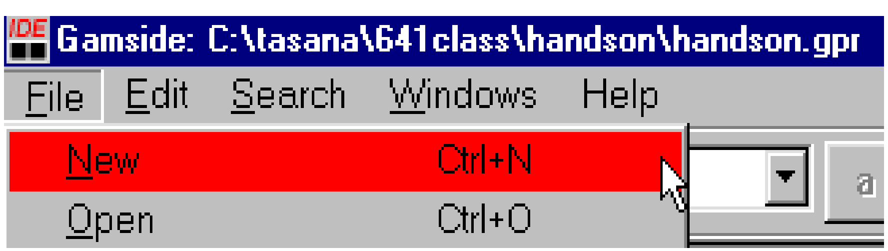
Step 2:

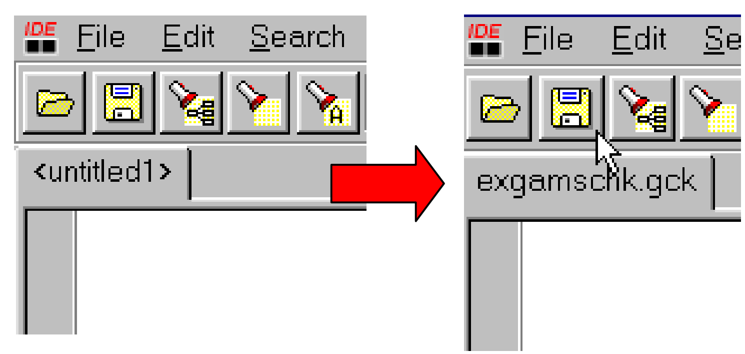
Create a new file with extension *.gck that has the same corresponding name as the program file. If your program file is called exgamschk.gms, then make a new file called exgamschk.gck





To create a new file, go to the FILE menu and use the NEW option. You will then get a file called untitled with an empty screen then save your program as exgamschk.gck using the file SAVE option.

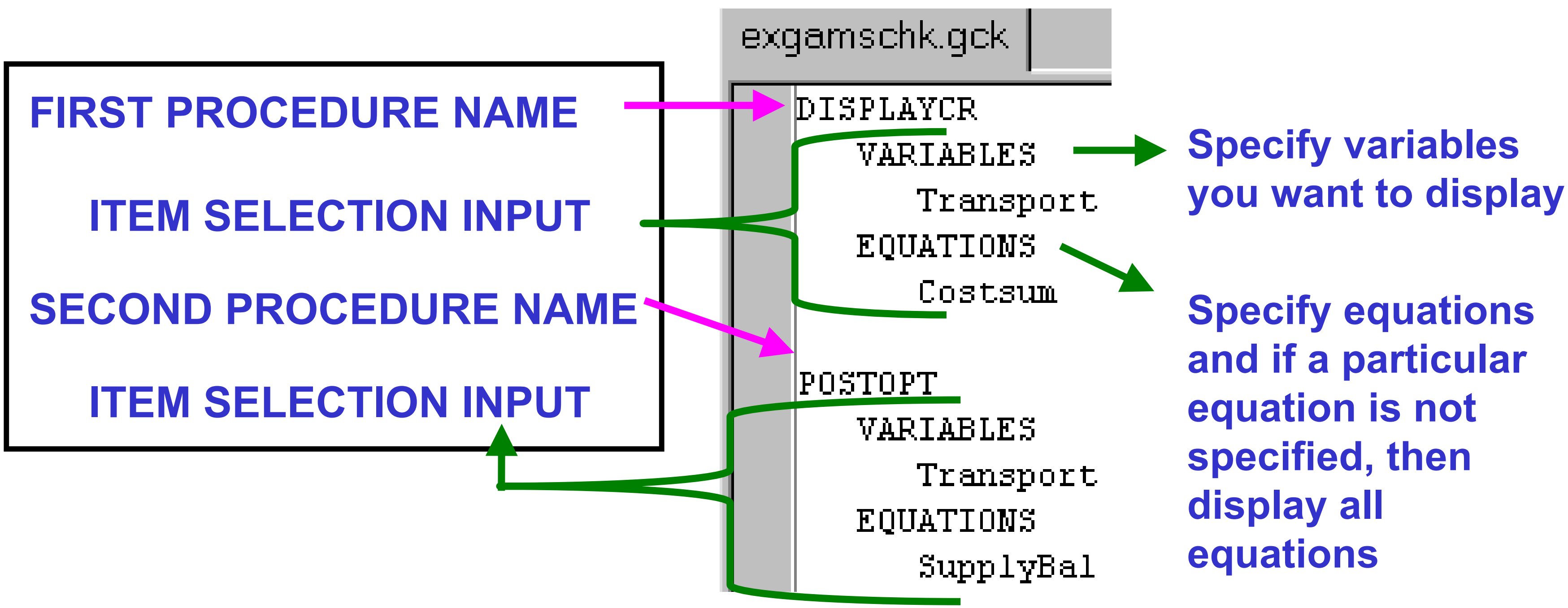




Selecting Procedures and Providing Inputs

GAMSCHK requires that the user indicate which procedures are to be employed. This is done in the GCK file. If a *.GCK file cannot be found, then it is assumed that the BLOCKPIC procedure is selected.

The general form of the *.gck file is:



Note that spaces and capitalization are ignored in this input.

Selecting Procedures and Providing Inputs

When variables or equations are selected after an item selection keyword, one can apply a number of input rules as follows:

1) If a variable or equation name is entered without any following parentheses, then all cases for that variable or equation are selected.

```
DISPLAYCR
   VARIABLES
      Transport
   EQUATIONS
      Costsum
```

2) If all elements from sets are selected, wild cards can be used.

```
DISPLAYCR
   VARIABLES
      Transport(Seattle,*)
```

Select cases where the first set element equals SEATTLE and any element from the second set

3) If a wild card is used to select items (e.g. Tr*), GAMS will select anything starting Tr.

```
DISPLAYCR
   VARIABLES
      Tr*
```

Including Comments in *.GCK File

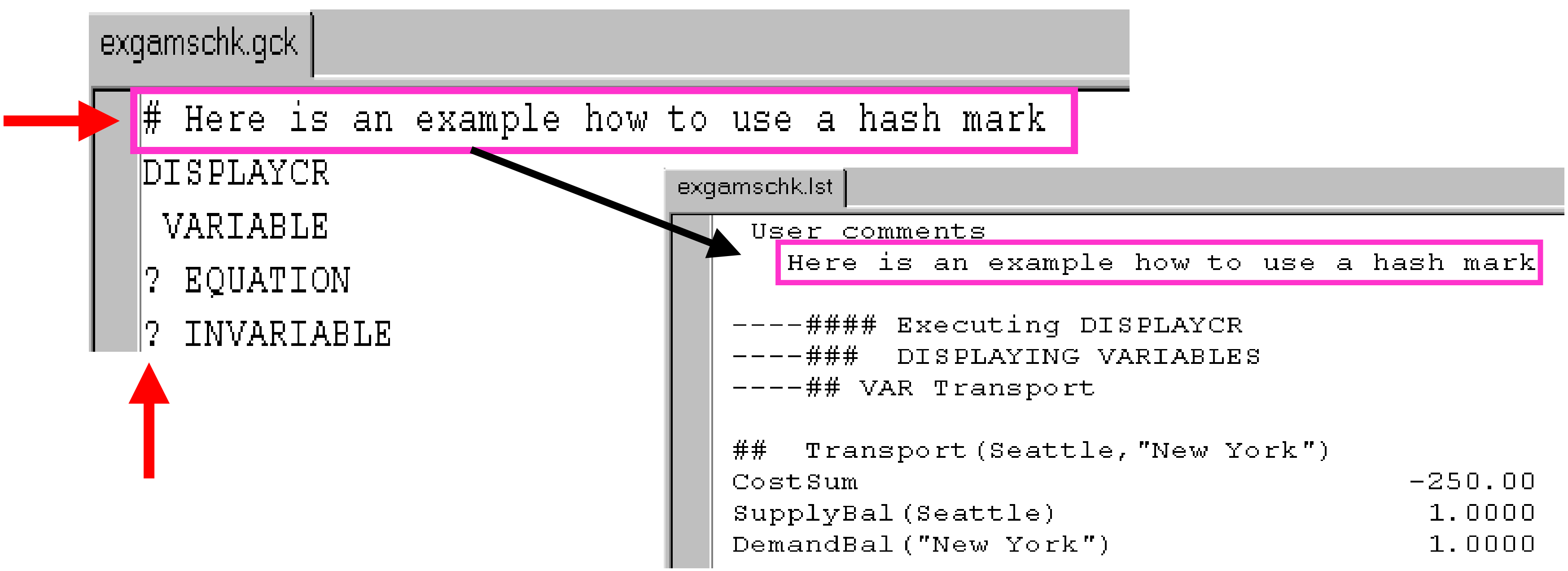
One can include comments in the *.GCK file through the use of

- a hash mark **#**

  comments that begin with a hash mark (#) are copied to the output when the program runs.

- a question mark **?**

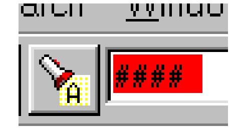
  comments which begin with a question mark **?** are simply overlooked.



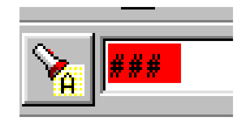
Output

The **exgamschk**.lst file contains the output on the following pages.

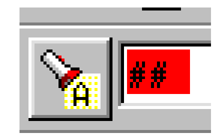
To find the major sections search for #### followed by a space



To find the second level sections => ### followed by a space



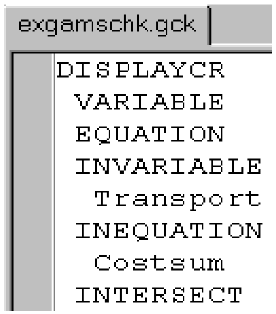
To find the lowest level sections => ## followed by a space




```
----#### Executing DISPLAYCR

----###  DISPLAYING VARIABLES

----## VAR Transport

##  Transport(Seattle,"New York")

CostSum                                -250.00
SupplyBal(Seattle)                      1.0000
DemandBal("New York")                   1.0000
```

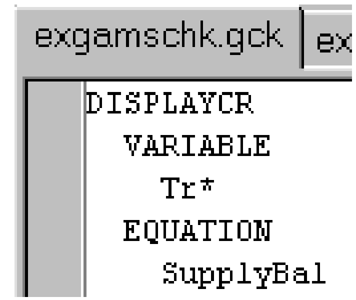
DISPLAYCR

DISPLAYCR produces output much like **LIMROW** and **LIMCOL**, but **DISPLAYCR** allows one to select specific items. The keywords are **VARIABLE, EQUATION, INVARIABLE, INEQUATION, INTERSECT.** The keywords may be used by themselves or followed with specific names


exgamschk.gck

```
DISPLAYCR
 VARIABLE
 EQUATION
 INVARIABLE
  Transport
 INEQUATION
  Costsum
 INTERSECT
```

Or, select specific items


exgamschk.gck ex

```
DISPLAYCR
  VARIABLE
    Tr*
  EQUATION
    SupplyBal
```

1) If VARIABLE and EQUATION are used, variable or equations are listed.

2) If INVARIABLE is used, equations where the selected variable fall are listed.

3) If INEQUATION is used, variables that fall in selected are listed.

4) If INTERSECT is used, coefficients which appear at intersections of selected variables and equations are listed.

GAMS CHECK – DISPLAYCR

```
----#### Executing DISPLAYCR

----###  DISPLAYING VARIABLES
----## VAR Transport

##  Transport(Seattle,"New York")
CostSum                         -250.00
SupplyBal(Seattle)               1.0000
DemandBal("New York")            1.0000

----###    DISPLAYING EQUATIONS
----## EQU SupplyBal

##  SupplyBal(Seattle)
Transport(Seattle,"New York")    1.0000
Transport(Seattle,Chicago)       1.0000
Transport(Seattle,Topeka)        1.0000
        =L=                      35.000
```

```
----###   DISPLAYING VARIABLES
           INTERSECTION MODE
----## VAR Transport

##  Transport(Seattle,"New York")
CostSum                         -250.00
SupplyBal(Seattle)               1.0000
DemandBal("New York")            1.0000

----###     DISPLAYING EQUATIONS
           INTERSECTION MODE
----## EQU SupplyBal

##  SupplyBal(Seattle)
Transport(Seattle,"New York")    1.0000
Transport(Seattle,Chicago)       1.0000
Transport(Seattle,Topeka)        1.0000
        =L=                      35.000
```

Blockpic

Why use BLOCKPIC?

1. BLOCKPIC is used to find GAMS coding errors in a model structure by looking at a whole summary of the model. Scaling can also be investigated

BLOCKPIC is good for looking at overall structure. PICTURE for individual item structure and DISPLAYCR for really getting Down to details

```
                 |         |
                 | T   T   |
                 | r   o   |
                 | a   t   |
                 | n   a   |
                 | s   l   |
                 | p   C   |
                 | o   o   |      R
                 | r   s   |      H
                 | t   t   |      S
----------------------------------
CostSum          | -   +   |   E  0
SupplyBal        | +       |   L  +
DemandBal        | +       |   G  +
----------------------------------
Variable Typ     | +   u
```

Blockpic

```
### B. Number of Coefficients by Block -- Strip    1
                |              |             C     #
                |    T     T   |             o
                |    r     o   |             e     o
                |    a     t   |             f     f
                |    n     a   |             f
                |    s     l   |
                |    p     C   |             C     E
                |    o     o   |        R    n     q
                |    r     s   |        H    t     n
                |    t     t   |        S    s     s
 -------------------------------------------------
 CostSum        |          1+  | E           1+    1
                |    6-        |             6-
 SupplyBal      |    6+        | L      2+   6+    2
                |              |
 DemandBal      |    6+        | G      3+   6+    3
                |              |
 -------------------------------------------------
 Coeff Cnts     |   12+    1+  |        5+  13+
                |    6-        |             6-
 # of  Vars     |    6     1   |
 Variable Typ   |   >=0   <0>
```

Blockpic

```
### C. Average Number of Coefficients by Column Block -- Strip    1
           |                                  C       #
           |      T       T                   f
           |      r       o                   s       o
           |      a       t                           f
           |      n       a                   P
           |      s       l                   e
           |      p       C                   r       E
           |      o       o           R       E       q
           |      r       s           H       q       n
           |      t       t           S       u       s
----------------------------------------------------------
CostSum    |              1+ | E              1+      1
           |      1-         |                6-
SupplyBal  |      1+         | L      2+      3+      2
           |                 |
DemandBal  |      1+         | G      3+      2+      3
----------------------------------------------------------
Cfs PerVar |      2+      1+ |
           |      1-         |
# of  Vars |      6       1
 Var Type  |    >=0     <0>
----------------------------------------------------------
```

Blockpic

D. Scaling - Maximum & Minimum Coefficients by Block -- Strip 1

		Transport	TotalCost	RHS MaxMin	Equ MaxMin
CostSum	Max	250	1		250
	Min	151	1		1
SupplyBal	Max	1		600	1
	Min	1		350	1
DemandBal	Max	1		325	1
	Min	1		275	1
Total Var	Max	250	1	600	
	Min	1	1	275	

BLOCKLIST

Why use BLOCKLIST ?

1. BLOCKLIST is used to look at the scaling problems. Good scaling improves the numerical accuracy of computer algorithms and reduces solution time. **Rule of thumb:** Scale when the matrix coefficient magnitudes differ in magnitude by more than 10^3 or 10^4.
2. The way scaling factors are utilized may be motivated by reference to the **homogeneity of units test**. The coefficients associated with any variable are homogeneous in terms of their denominator units. Constraints, however, possess homogeneity of numerator units.

Blocklist

exgamschk.gck

BLOCKLIST

Look at scaling of the problem

```
----#### Executing BLOCKLIST

 ----### List of Variable Block Characteristics

 Note Max and Min do not include Objective Row

 Variable    Sign   Numb  Numb  Pos   Neg   Nonl   Maximum    Minimum
   Block     Res    Vars  Nonl  Coef  Coef  Coef   Absolute   Absolute
 Transport   >=0      6     0    12     6     0     1.000      1.000
 TotalCost   <0>      1     0     1     0     0    0.0000E+000.0000E+00


 ----### List of Equation Block Characteristics

 Note Max and Min do not include RHS and Objective variable

 Equation    Type   Numb  Numb  Pos   Neg   Nonl  Pos   Neg   Maximum   Minimum
   Block     Res    Eqns  Nonl  Coef  Coef  Coef  RHS   RHS   Absolute   Absolute
 CostSum     =E=      1     0     1     6     0     0     0    250.0      151.0
 SupplyBal   =L=      2     0     6     0     0     2     0    1.000      1.000
 DemandBal   =G=      3     0     6     0     0     3     0    1.000      1.000
```

ANALYSIS

ANALYSIS is used to analyze the structure of all variables and equations. Information from **ANALYSIS** is used to define if individual variables or equations in the model have specification errors which lead to redundancy, zero variable values, infeasibility, unboundedness, or obvious constraint redundancy in linear programs.

For example, analysis tells you if, a variable appears which has a positive return in the objective function, but no resource using coefficients in the constraints- obviously unbounded.

For example, it will also complain if an equation appears with a negative right hand side and all positives on the left hand side - obviously infeasible model.

$$
\begin{array}{lrcr}
\text{Max} & 3X + 2Y + 4Z & & \\
 & 2X + 3Y & \leq & -5 \\
 & -2X - Y + Z & \geq & 4 \\
 & X, \ Y & \geq & 0
\end{array}
$$

Rule Examples

Cases where the model must have an infeasible solution

$$
\begin{array}{llllll}
Max & \sum_j c_j X_j & & & & \\
s.t. & \sum_j a_{ij} X_j & \leq & b_i & \text{for all} & i \\
 & \sum_j e_{nj} X_j & = & d_n & \text{for all} & n \\
 & \sum_j f_{mj} X_j & \geq & g_m & \text{for all} & m \\
 & X_j & \geq & 0 & \text{for all} & j
\end{array}
$$

$$
\begin{array}{lrl}
\text{Max} & 3X + 2Y & \\
 & 2X + 3Y & \leq -5 \\
 & -2X - Y & \geq 4 \\
 & X, Y & \geq 0
\end{array}
$$

$b_i < 0$ and $a_{ij} \geq 0$ for all j => row i will not allow a feasible solution

$d_n < 0$ and $e_{nj} \geq 0$ for all j => row n will not allow a feasible solution

$d_n > 0$ and $e_{nj} \leq 0$ for all j => row n will not allow a feasible solution

$g_m > 0$ and $f_{mj} \leq 0$ for all j => row m will not allow a feasible solution

Rule Examples

Cases where certain variables in the model must equal zero

$$
\begin{array}{llllll}
Max & \sum_j c_j X_j & & & & \\
s.t. & \sum_j a_{ij} X_j & \leq & b_i & for\ all & i \\
 & \sum_j e_{nj} X_j & = & d_n & for\ all & n \\
 & \sum_j f_{mj} X_j & \geq & g_m & for\ all & m \\
 & X_j & \geq & 0 & for\ all & j
\end{array}
$$

$$
\begin{array}{lll}
\text{Max} & 3X + 2Y & \\
 & +1X + 3Y & \leq 0 \\
 & -2X - 1Y & \geq 0 \\
 & X, \quad Y & \geq 0
\end{array}
$$

$b_i = 0$ and $a_{ij} \geq 0$ for all $j \Rightarrow$ all X_j's with a_{ij} NE 0 in row i will be zero

$d_n = 0$ and $e_{nj} \geq 0$ for all $j \Rightarrow$ all X_j's with e_{nj} NE 0 in row n will be zero

$d_n = 0$ and $e_{nj} \leq 0$ for all $j \Rightarrow$ all X_j's with e_{nj} NE 0 in row n will be zero

$g_m = 0$ and $f_{mj} \leq 0$ for all $j \Rightarrow$ all X_j's with f_{mj} NE 0 in row m will be zero

Rule Examples

Cases where certain constraints are obviously redundant

$$
\begin{array}{llllll}
Max & \sum_j c_j X_j & & & & \\
s.t. & \sum_j a_{ij} X_j & \le & b_i & \text{for all} & i \\
 & \sum_j e_{nj} X_j & = & d_n & \text{for all} & n \\
 & \sum_j f_{mj} X_j & \ge & g_m & \text{for all} & m \\
 & X_j & \ge & 0 & \text{for all} & j
\end{array}
$$

$$
\begin{array}{llll}
\text{Max} & 3X + 2Y & & \\
 & -2X - 3Y & \le & 5 \\
 & 2X + 1Y & \ge & -4 \\
 & -X & \le & 10 \\
 & -Y & \le & 10 \\
 & X, \quad Y & \ge & 0
\end{array}
$$

$b_i \ge 0$ and $a_{ij} \le 0$ for all j means row i is redundant

$g_m \le 0$ and $f_{mj} \ge 0$ for all j means row m is redundant

Rule Examples

Cases where certain variables will be unbounded

$$
\begin{array}{llllll}
Max & \sum_j c_j X_j & & & & \\
s.t. & \sum_j a_{ij} X_j & \le & b_i & for\ all & i \\
 & \sum_j e_{nj} X_j & = & d_n & for\ all & n \\
 & \sum_j f_{mj} X_j & \ge & g_m & for\ all & m \\
 & X_j & \ge & 0 & for\ all & j
\end{array}
$$

$$
\begin{array}{lrcl}
\text{Max} & 3X + 2Y & & \\
 & -X - 3Y & \le & 5 \\
 & 2X + Y & \ge & -1 \\
 & X,\ Y & \ge & 0
\end{array}
$$

$c_j > 0$ and $a_{ij} \le 0$ or $e_{nj} = 0$ and $f_{mj} \ge 0$ for all i, n, and m means variable j will never be nonzero

Rule Examples

Cases where certain variables will be zero at solution

$$
\begin{array}{lllll}
Max & \sum_j c_j X_j & & & \\
s.t. & \sum_j a_{ij} X_j & \leq & b_i & \text{for all } i \\
 & \sum_j e_{nj} X_j & = & d_n & \text{for all } n \\
 & \sum_j f_{mj} X_j & \geq & g_m & \text{for all } m \\
 & X_j & \geq & 0 & \text{for all } j
\end{array}
$$

$$
\begin{array}{l}
\text{Max } -3X - 2Y \\
+ X + 3Y \leq 5 \\
-2X - Y \geq -4 \\
X, \; Y \geq 0
\end{array}
$$

$c_j < 0$ and $a_{ij} \geq 0$ or $e_{nj} = 0$ and $f_{mj} \leq 0$ for all i, n, and m means variable j will never be nonzero

ANALYSIS

exgamschk.gck

ANALYSIS

```
**** Warning  These variables will equal zero
  because they have a zero lower bound
  an undesirable object function coefficient
  all 0 or - coefficients in the =G= rows
  all 0 or + coefficients in the =L= rows
  and no coefficients in the =E= rows
##  movecrops
```

	profit	feedcattle	movecrops	growcrops	sellcrops	landrent		RHS
profitacct	+	-	+	+	m	+	E	0
croponhand		+	+	m	+		L	0
Land		+		+		-	L	+
mincattle		+					G	+
rentalLand						+	L	-
Variable Typ	u	+	+	+	+	+		

```
**** ERROR This =L= constr. causes an infeasible model
  since the nonnegative variables present
           have only 0 or + coefficients
       the nonpositive variables present
           have only 0 or - coefficents
       the unrestricted variables
           have only zero coefficients
       and the RHS is negative
 ##  rentalLand
```

Picture

Why use PICTURE? PICTURE permit investigation and verification of the structure of equations and variables.

1. Look at interrelationships between items
   - how coefficients for a variable appear across equations?
   - What variables appear in an equation?
   - How some variables balance against other variables in equations?
   - How signs are distributed?
2. Look at magnitude, sign and location of coefficients
3. Avoid immense output from using LIMROW/LIMCOL or DISPLAYCR

Picture


```
----#### Executing PICTURE

 ### PICTURE -  COEFFICIENT CODES

    LOWER BOUND    CODE     UPPER BOUND
     (INCLUSIVE)              (LESS THAN)
      1000.00000       G       +INFINITY
       100.00000       F        1000.00000
        10.00000       E         100.00000
         1.00000       D          10.00000
         1.00000       C           1.00000
         0.50000       B           1.00000
         0.00000       A           0.50000
         0.00000       0           0.00000
        -0.50000       1           0.00000
        -1.00000       2          -0.50000
        -1.00000       3          -1.00000
       -10.00000       4          -1.00000
      -100.00000       5         -10.00000
     -1000.00000       6        -100.00000
     -INFINITY         7       -1000.00000
```

Picture

```
           |                   R
           | T T T T T T T     H
           | r r r r r r o     S     P     N
           | a a a a a a t           O     E     R
           | n n n n n n a     C     S     G     O
           | s s s s s s l     O     I A   A A   W
           | p p p p p p C     E     T I   T I   C
           | o o o o o o o     F     I J     J   N
           | r r r r r r s     F     V ,   V ,   T
           | t t t t t t t     S     E S   E S   S
           |
           | 1 2 3 4 5 6 1
            --------------
  CostSum 1| 6 6 6 6 6 6 C   = 0       1     6     7
SupplyBal 1| C C C           < F       3     0     3
SupplyBal 2|       C C C     < F       3     0     3
DemandBal 1| C     C         > F       2     0     2
DemandBal 2|   C     C       > F       2     0     2
DemandBal 3|     C     C     > F       2     0     2
            --------------
LOWER BND  | 0 0 0 0 0 0 -
UPPER BND  | + + + + + + +
            --------------
```

Picture

```
  ----### Dictionary of Variables

Transport         1:  Transport(Seattle,"New York")
Transport         2:  Transport(Seattle,Chicago)
Transport         3:  Transport(Seattle,Topeka)
Transport         4:  Transport("San Diego","New York")
Transport         5:  Transport("San Diego",Chicago)
Transport         6:  Transport("San Diego",Topeka)
TotalCost         1:  TotalCost


  ----### Dictionary of Equations

CostSum           1:  CostSum
SupplyBal         1:  SupplyBal(Seattle)
SupplyBal         2:  SupplyBal("San Diego")
DemandBal         1:  DemandBal("New York")
DemandBal         2:  DemandBal(Chicago)
DemandBal         3:  DemandBal(Topeka)
```

MATCHIT

MATCHIT is used to retrieve names and characteristics of selected variables or equations. The characteristics reported tell whether the items are nonlinear as well as reporting scaling characteristics and counts of the coefficients. The keywords for this procedure are VARIABLE, EQUATION, LISTVARIABLE, LISTEQUATION and again can be followed by selections

If VARIABLE is used, MATCHIT summarizes the characteristics of either all variables or those named

If EQUATION is used, MATCHIT summarizes the characteristics of either all equations or those named.

exgamschk.gck

```
MATCHIT
   VARIABLE
   EQUATION
   LISTVARIABLE
   LISTEQUATION
```

----### Variable Request	Numb Varia	Numb Nonln	Total Coef	Pos Coef	Neg Coef	Nonln Coef
All Variables	7	0	19	13	6	0

----### Equation Request	Numb Equat	Numb Nonln	Total Coef	Pos Coef	Neg Coef	Nonln Coef
All Equations	6	0	24	18	6	0

MATCHIT

If LISTVARIABLE or LISTEQUATION are used, MATCHIT summarizes the characteristics of individual variables or equations including

is it nonlinear?; how many total coefficients it has?; count of positive, negative, and nonlinear coefficients, minimum and maximum absolute values of coefficients

```
----#### Executing MATCHIT

Note Max and Min do not include Obj row coef

                                    Is   Tot  Pos  Neg  Nln   Minimum    Maximum
 ----### Requested Variables        Non  Cof  Cof  Cof  Cof   Absolute   Absolute


----## VAR Transport

Transport(Seattle,"New York")       0    3    2    1    0    1.000      1.000
Transport(Seattle,Chicago)          0    3    2    1    0    1.000      1.000
Transport(Seattle,Topeka)           0    3    2    1    0    1.000      1.000
Transport("San Diego","New York")   0    3    2    1    0    1.000      1.000
Transport("San Diego",Chicago)      0    3    2    1    0    1.000      1.000
Transport("San Diego",Topeka)       0    3    2    1    0    1.000      1.000
```

Postopt

Why use POSTOPT?

It is used to investigate a misbehaving model yielding an unrealistic, infeasible, or unbounded solution.

In general, models are infeasible or unbounded because they have components within them that interact to make an infeasible or unbounded solution.

The first thing we should do is to identify the offending components identifying which ones to look at in the total model framework differentiating them from all the other the elements of the model which may not be in the least involved with the infeasibility or unboundedness.

Infeasible Example

```
VARIABLE
 ObjMax      Max objective function;
POSITIVE VARIABLES
 Corn        Corn production
 Soybeans    Soybeans production  ;
EQUATIONS
 ObjBal      Objective function
 LandBal     Land constraint
 LaborBal    Labor constraint
 MinRequirement Min land;

ObjBal..
 ObjMax =E= 50*Corn + 50*Soybeans;
LandBal..
 Corn + Soybeans    =L=  50;
LaborBal..
 50*Corn + Soybeans =L=  65;
MinRequirement..
 Corn =G=  20;

MODEL ExInfes /all/;

SOLVE ExInfes using lp maximizing ObjMax;
```

The Labor and MinRequirement constraints cannot be simultaneously satisfied. WHY? Is it because …

The 20 is too large a lower bound on Corn, or

The coefficient of 50 for Corn Labor constraint is too large, or

The Soybeans labor coefficient should have been large and negative, or

The 65 RHS on Labor is too small.

Furthermore the first constraint is irrelevant to the infeasibility.

So how do we find which constraints to examine?

Finding Infeasible Problem

When facing an infeasible problem, how to discover the cause?

1. First, conduct a PRE-solution check to find any model formulation defects using GAMSCHECK procedures discussed previously (e.g. PICTURE or ANALYSIS). After that, if the model is still infeasible, then what?

2. Next, use artificial variables (ART) by including the ART to the model formulation.

   a. ART has a large negative objective function coefficient (maximization problem) and positive in a single constraint.

   b. ART permits infeasible solutions to appear feasible.

   c. Nonzero ART causes a large negative objective function value and large shadow prices (C_B's associated w/ ART in the $C_B B^{-1}$ are large). So, one should pay attention to constraints having large shadow prices or reduced costs.

Using Artificial to Find Infeasible Problem

Why does the objective value have a large negative value?

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

     MODEL   ExInfes              OBJECTIVE  ObjMax
     TYPE    LP                   DIRECTION  MAXIMIZE
     SOLVER  BDMLP                FROM LINE  36

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE         -18699935.0000
```

```
EXIT -- OPTIMAL SOLUTION FOUND.

                     LOWER     LEVEL     UPPER    MARGINAL
---- EQU ObjBal        .         .         .         EPS
---- EQU LandBal     -INF      1.300    50.000        .
---- EQU LaborBal    -INF     65.000    65.000  20001.000
---- EQU MinRequir   20.000   20.000     +INF   -1.000E+6

                     LOWER     LEVEL     UPPER    MARGINAL
---- VAR ObjMax      -INF   -1.870E+7    +INF        .
---- VAR Corn          .       1.300     +INF        .
---- VAR Soybeans      .         .       +INF   -1.995E+4
---- VAR Artificial    .      18.700     +INF        .
```

```
VARIABLE
 ObjMax       Max objective function;
POSITIVE VARIABLES
 Corn         Corn production
 Soybeans     Soybeans production
 Artificial   Artificial variable   ;

EQUATIONS
 ObjBal       Objective function
 LandBal      Land constraint
 LaborBal     Labor constraint
 MinRequirement Min land;

ObjBal..
 ObjMax =E=
 50*Corn + 50*Soybeans -1000000 * Artificial;
LandBal..
 Corn + Soybeans    =L=  50;
LaborBal..
 50*Corn + Soybeans =L=  65;
MinRequirement..
 Corn + Artificial  =G=  20;

MODEL ExInfes /all/;
SOLVE ExInfes using lp maximizing ObjMax;
```

Why are these values big?

Finding Unrealistic Solutions

One could use the techniques called row summing which is used to breakdown a constraint by row (Ax = b). This technique is implemented in GAMSChk.

A particular equation is not specified, then display all equations

infe.gms | infe.gck | infe.lst

```
POSTOPT
  EQUATION
```

infe.gms | infe.gck | infe.lst

```
POSTOPT
  EQUATION
    LaborBal
```

Specify equations

Row Sum layout for row i

Variable Names	Coefficients from data	Solution values	Calculated product
X_1	a_{i1}	X_1^*	$a_{i1}X_1^*$
X_2	a_{i2}	X_2^*	$a_{i2}X_2^*$
...	...	...	...
X_n	a_{in}	X_n^*	$a_{in}X_n^*$
Sum	- -	- -	$\Sigma_j a_{ij} X_j^*$
RHS	- -	- -	b
Slack	- -	- -	$b-\Sigma_j a_{ij} X_j^*$

```
----###   ROW SUMMING EQUATIONS
----## EQU ObjBal
##  ObjBal
  VAR                    Aij          Xj          Aij*Xj
ObjMax                1.0000      65.000          65.000
Corn                 -50.000      1.3000         -65.000
Soybeans             -50.000  0.00000E+00    0.00000E+00
   =E=                                             =E=
RHS COEFF                                    0.00000E+00
SHADOW PRICE                                 0.00000E+00

----## EQU MinRequirement
##  MinRequirement
  VAR                    Aij          Xj          Aij*Xj
Corn                  1.0000      1.3000          1.3000
   =G=                                             =G=
RHS COEFF                                         20.000
SURPLUS EQUALS                                   -18.700
SHADOW PRICE                                 0.00000E+00
```

Finding Unrealistic Solution Problems

Steps to Use Row Summing

- Find a variable or slack with an unreasonable value
- Choose a constraint where this variable or slack appears
- Examine the allocation calculations to find other unrealistic variable level values or aij/rhs data errors which balance off allowing the unreasonable value for the originally sought item.
- If no other bad variable or data are found, then examine another constraint.
- If another variable is found to have a bad level, then examine another constraint into which it falls
- If bad data are found repair the model

Let use the example from Table 17.14 in McCarl and Spreen book

to illustrate the use of Row Summing

Table 17.14. Row Summing Example

Row	Buy Misc.	Sell Corn	Sell Soyb.	Sell Pork	Prod Corn	Prod Soyb.	Prod Hogs	RHS
Objective Func	-1	2.5	6	0.5	-75	-50		MAX
Land Available					1	1		≤ 600
Pork Balance				1			-150	≤ 0
Soybean Bal			1			-50		≤ 0
Corn Balance		1			-1200		20	≤ 20
Misc. Inp. Bal.	-1				125	50	20	≤ 0

Finding Unrealistic Problem

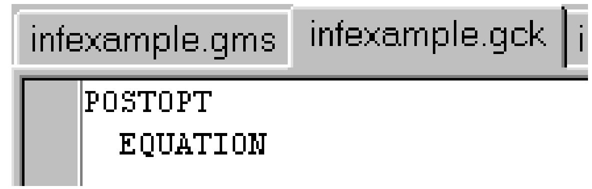
```
SET     Product    Production    /Corn,Soybean,Pork/ ;
ALIAS   (Product,Product1);
SET     mapProduct(Product,Product1)
         /Corn.(Corn,Soybean,Pork)
          Soybean.(Corn,Soybean,Pork)
          Pork.(Corn,Soybean,Pork)/   ;
PARAMETER
  Price(Product)    Selling prices
   /Corn              2.5
    Soybean           6
    Pork              0.5
   /
  Cost(Product)    Produciton costs
   /Corn              75
    Soybean           50
    Pork               0
   /
 OnHand(Product)   Production on hand
   /Corn              20
    Soybean            0
    Pork               0
   /
 BuyAij(Product)   Production on hand
   /Corn              125
    Soybean            50
    Pork               20
   /
 UseLand           If 1 Use land otherwise not use land
   /Corn              1
    Soybean           1
  TABLE  ProdAij(Product,Product1)
                        Corn     Soybean    Pork
       Corn             1200                -20
       Soybean                       50
       Pork                                 150    ;
```

```
  VARIABLE
   Profit                    Profit ;
  POSITIVE VARIABLE
   Sell(Product)             Sales
   Production(Product)       Production
   BuyMisc                   Misc ;
  EQUATIONS
   Objt                      Objective function
   LandBal                   Land balance
   ProductBal(Product)       Production balance
   BuyBal                    Buy balance    ;

Objt..
   Profit
   =E=
   SUM(Product,Price(Product)*Sell(Product))
 - SUM(Product,Cost(Product)*Production(Product))
 - BuyMisc   ;

LandBal..
   SUM(Product$UseLand(Product),Production(Product)) =L= 600;

ProductBal(Product)..
   Sell(Product)
    =L=
   SUM(mapProduct(Product,Product1),
         ProdAij(Product,Product1)*Production(Product1))
 + OnHand(Product);

BuyBal..
   SUM(Product,BuyAij(Product)*Production(Product))
   =L= BuyMisc;

MODEL  InfEx /ALL/   ;
OPTION LP = GAMSCHK;
SOLVE  InfEx using lp maximizing Profit;
```

```
##   ProductBal(Pork)
  VAR                        Aij         Xj         Aij*Xj
Sell(Pork)                 1.0000     0.54002E+07 0.54002E+07
Production(Pork)          -150.00     36001.      -0.54002E+07
   =L=                                                =L=
RHS COEFF                                         0.00000E+00
SLACK EQUALS                                      0.00000E+00
SHADOW PRICE                                      0.50000
```

```
##   BuyBal
  VAR                        Aij         Xj         Aij*Xj
Production(Corn)           125.00     600.00      75000.
Production(Soybean)        50.000     0.00000E+00 0.00000E+00
Production(Pork)           20.000     36001.      0.72002E+06
BuyMisc                   -1.0000     0.79502E+06-0.79502E+06
   =L=                                               =L=
RHS COEFF                                        0.00000E+00
SLACK EQUALS                                     0.00000E+00
SHADOW PRICE                                     1.0000
```

```
##   ProductBal(Corn)
  VAR                        Aij         Xj         Aij*Xj
Sell(Corn)                 1.0000     0.00000E+00 0.00000E+00
Production(Corn)          -1200.0     600.00      -0.72000E+06
Production(Pork)           20.000     36001.      0.72002E+06
   =L=                                                =L=
RHS COEFF                                         20.000
SLACK EQUALS                                      0.00000E+00
SHADOW PRICE                                      2.7500
```



Is Pork production reasonable (36001 hogs)?

Then look at resource used by Pork production. Let's start with investigating the BuyBal constraint.

What is wrong here?

Next, investigate the corn demand-supply balance.

We find that 36001 hogs require 720,010 bushels of corn. Is it reasonable? So, what is the problem here?

Unbounded Example

```
VARIABLE
 ObjMax      Max objective function;
POSITIVE VARIABLES
 Corn        Corn production
 Water       Water
 Cotton      Dryland Cotton production ;
EQUATIONS
 ObjBal      Objective function
 WaterBal    Water constraint
 MaxLand     Max cotton land ;

ObjBal..
 ObjMax =E= 3*Corn - Water + 2*Cotton ;
WaterBal..
 Corn - Water  =L=  0;
MaxLand..
 Cotton =L= 20;

MODEL ExUnbound /all/;

SOLVE ExUnbound using lp maximizing ObjMax;
```

The Corn and Water variables can be raised to infinity while still making money. So, what is wrong with this structure? Is it because …

1. **The omission on constraints for Corn and Water, or**
2. **The omission of some sort of decreasing marginal revenue or increasing cost function that affects Corn and Water, or**

Which of these reasons are the cause of unbounded?

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

     MODEL   ExUnbound            OBJECTIVE  ObjMax
     TYPE    LP                   DIRECTION  MAXIMIZE
     SOLVER  BDMLP                FROM LINE  35

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      3 UNBOUNDED
**** OBJECTIVE VALUE               40.0000
```

GAMS MODEL STATUS will report what solution looks like (optimal, infeasible, unbounded, etc.). In this case, it is unbounded.

Finding Unbounded Problem

When facing an unbounded problem, how to discover the cause?

1. First, conduct a PRE-solution check to find any model formulation defects using GAMSCHECK procedures discussed previously .
2. Next, add large bounds to all variables which increase objective.

   In a maximization context

   a. Non negative with positive objective function coefficients where we use large upper bounds

```
Corn.UP    = 1000000;
Water.UP   = 1000000;
Cotton.UP = 1000000;
MODEL ExUnbound /all/;
```

   b. Non positive with negative objective function coefficients which need large negative lower bounds

   c. Unrestricted with positive objective function coefficients where we need a large upper bound

   d. Unrestricted with negative objective function coefficients where we need large negative lower bounds

   Solve the resultant model. If any of the imposed large bounds are binding, then find set of all variables with solution levels which are unrealistically large in absolute value.

Finding Unbounded Problem

Steps to Use Budgeting

- Choose a variable to budget which exhibits a bad reduced cost in the solution information or which uses resources with bad shadow prices
- Examine how things balance out then examine rows where things look bad in terms of the contained shadow prices and a_{ij}'s to find either unrealistically high shadow prices or data errors
- If an excessively high shadow price has been found then budget other basic variables which use the resource involved
- If an error in the a_{ij}'s is found which is causing the distortion then repair the model

Let use the example from Table 17.9 in McCarl and Spreen book to illustrate the use of Budgeting

Table 17.9. Tableau of Budgeting Example

Row	Buy Misc.	Sell Corn	Sell Soyb.	Sell Pork	Prod Corn	Prod Soyb.	Prod Hogs	RHS
Objective Func	-1	2.5	6	0.5	-75	-50		MAX
Land Available					1	1		≤ 600
Pork Balance				1			-1000	≤ 0
Soybean Bal			1			-50		≤ 0
Corn Balance		1			-120		20	≤ 0
Misc. Inp. Bal.	-1				125	50	20	≤ 0

Table 17.15. Optimal Solution to Row Summing Example

Variable	Value	Reduced Cost	Equation	Level	Shadow Price
Buy Misc. Input	795,020	0	Land Available	0	3,100
Sell Corn	0	0.25	Pork Balance	0	0.5
Sell Soybeans	0	0	Soybean Balance	0	6.00
Sell Pork	5,400,150	0	Corn Balance	0	2.75
Produce Corn	600	0	Misc. Input Balance	0	1.00
Produce Soybeans	0	2,480.00			
Produce Hogs	36,001	0			

```
##  Production(Soybean)
  SOLUTION VALUE                 0.000000E+00
  EQN                                 Aij          Ui        Aij*Ui
Objt                               50.000      1.0000        50.000
LandBal                            1.0000      2680.0        2680.0
ProductBal(Soybean)               -50.000      6.0000       -300.00
BuyBal                             50.000      1.0000        50.000
TRUE REDUCED COST                                            2480.0
```

```
infexample.gms   infexample.gck
POSTOPT
  VARIABLE
    Production(soybean)
```

Why is soybean's reduced cost so high? .. because of high land values. Then, why is land so valuable?

```
##  LandBal
  VAR                        Aij          Xj           Aij*Xj
Production(Corn)           1.0000      600.00          600.00
Production(Soybean)        1.0000      0.00000E+00  0.00000E+00
   =L=                                                 =L=
RHS COEFF                                              600.00
SLACK EQUALS                                      0.00000E+00
SHADOW PRICE                                           2680.0
```

```
infexample.gms   infexample.gck
POSTOPT
  EQUATION
    LandBal
```

Only Corn uses land so budgeting Corn?

```
##  Production(Corn)
  SOLUTION VALUE          600.000
  EQN                        Aij          Ui        Aij*Ui
Objt                      75.000      1.0000        75.000
LandBal                   1.0000      2680.0        2680.0
ProductBal(Corn)         -120.00      24.000        2880.0
BuyBal                    125.00      1.0000        125.00
TRUE REDUCED COST                              0.00000E+00
```

```
infexample.gms   infexample.gck
POSTOPT
  VARIABLE
    Production(corn)
```

Why is the corn shadow price so high?

Only Hogs consume Corn so budgeting Hogs production.

```
##  Production(Pork)
  SOLUTION VALUE          3601.00
  EQN                        Aij          Ui        Aij*Ui
ProductBal(Corn)          20.000      24.000        480.00
ProductBal(Pork)         -1000.0      0.50000      -500.00
BuyBal                    20.000      1.0000        20.000
TRUE REDUCED COST                              0.00000E+00
```

```
infexample.gms   infexample.gck
POSTOPT
  VARIABLE
    Production(pork)
```

NONOPT

Why use NONOPT?

It is used to help find the set associated with infeasibility problems. In particular, when running NONOPT on a solved model the reduced costs and shadow prices which are larger in absolute value than a tolerance given by 10 to the margfilt parameter set in the GAMSCHK option file.

Here is the GAMSCHK option file called GAMSCHK.OPT.


gamschk.opt

```
LEVELFILT   7
MARGFILT    7
```

This will cause the reporting of all marginals and levels which are greater in absolute value than 10^{7}.

LEVELFILT	**Numerical value of exponent on “unbounded levels”**
MARGFILT	**Numerical value of exponent on “infeasible marginals”**

NONOPT

NONOPT may be followed by optional keywords *IDENTIFY* or *VERBOSE*.

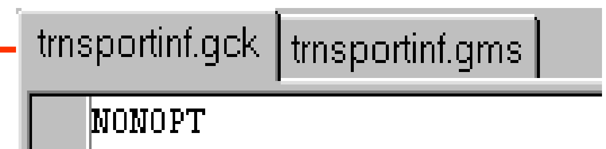
***IDENTIFY* keyword causes GAMSCHK to report potential unbounded variables and/or infeasible equations.**

***VERBOSE* causes full budgets and row summing as done by the POSTOPT procedure on infeasible equations, and/or variables as well as unbounded variables and/or equations.**

No keyword is found and the model solution is not optimal then the nonoptimal equations, infeasible equations and/or nonoptimal variables are automatically listed.

NONOPT

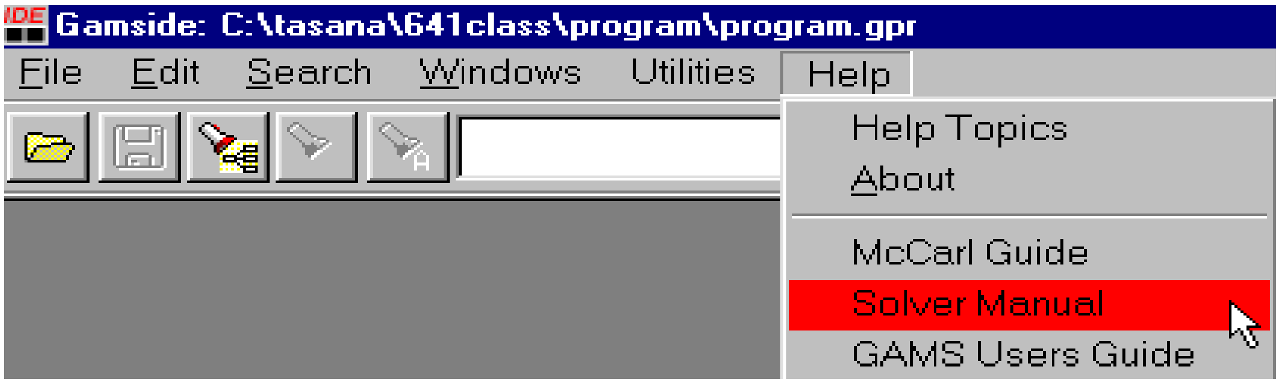
```
               S O L V E      S U M M A R Y

     MODEL    Tranport                  OBJECTIVE  TotalCost
     TYPE     LP                        DIRECTION  MINIMIZE
     SOLVER   CPLEX                     FROM LINE  70

**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      4 INFEASIBLE
**** OBJECTIVE VALUE             285.0000

 RESOURCE USAGE, LIMIT          0.000      1000.000
 ITERATION COUNT, LIMIT         1          10000
```




```
----#### Executing NONOPT


  ----###   LISTING INFEASIBLE EQUATIONS



      Supplybal(Seattle)
        Slack -285.00000     Dual -1.0000000
        RHS     350.00000
```

GAMSCHK – content description

See GAMSCHK user documentation for GCK file content description



The Solver Manuals - Table of Contents

Solver Manuals	
Introduction	Using Solver Specific Options
BDMLP	LP solver that comes with any GAMS system
CONOPT	Large scale NLP solver from ARKI Consulting and Development
CPLEX	High-performance LP/MIP solver from Ilog
Other Solver Documents	
BARON	Branch-And-Reduce Optimization Navigator for proven global solutions from The Optimization Firm
GAMSBAS	A Program for Saving an Advanced Basis for GAMS
GAMSCHK	A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved using GAMS
MPSGE	Modeling Environment for CGE models from University of Colorado at Boulder

Abridged GAMSCHK USER DOCUMENTATION

Version 1.1

A System for Examining the Structure and Solution Properties of Linear Programming Problems Solved using GAMS

by

Bruce A. McCarl
Professor
Department of Agricultural Economics
Texas A&M University

GAMS CHECK

GAMSCHK USER DOCUMENTATION

Hands On

(handson8.gms)

Learning Objectives:

1. Learn about GAMSCHK analysis

Please open handson3.gms and save it as handson8.gms. Then please run handson8.gms with GAMSCHK analysis using commands DISPLAYCR, PICTURE, BLOCKPIC, and POSTOPT.

References

McCarl, B. A. Basic GAMS class.
(http://ageco.tamu.edu/faculty/mccarl/mccarl.htm).