

# **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

(409) 845-7504 (fax)  
mccarl@tamu.edu

© Bruce A. McCarl  
June 25, 1998

## GAMSCHK USER DOCUMENTATION

General Notes on Package Usage .....	1
Selecting a Procedure and Providing Input -- the *.GCK File .....	2
The *.GCK file: General Notes on Item Selection .....	3
Procedure Output .....	5
Nonlinear Terms .....	6
Entering Comments in the *.GCK File .....	6
Controlling Page Width in the *.GCK File .....	6
Running Multiple Procedures .....	6
Use of the Procedures .....	7
DISPLAYCR .....	7
MATCHIT .....	9
ANALYSIS .....	11
BLOCKLIST .....	11
BLOCKPIC .....	12
PICTURE .....	13
POSTOPT .....	14
ADVISORY .....	15
NONOPT .....	16
Options File .....	17
Solver Choice Options .....	17
When Should I Use <i>SOLVE</i> or <i>NOSOLVE</i> .....	17
Control of Number of Variable and Row Selections Allowed .....	18
Scaling .....	18
NONOPT Filters .....	18
Example Options File .....	19
Solver Options File .....	19
Known Bugs .....	19
References .....	21
Appendix A: Reserved Names .....	26
Appendix B: GAMSCHK One Page Summary .....	27

## LIST OF TABLES

Table 1.	Conditions under which a modeler should be advised of potential difficulty for equations without nonlinear terms. . . . .	22
Table 2.	Conditions under which a modeler should be warned about variables in a maximization problem. . . . .	23
Table 3.	Conditions When Model Elements Could be Unbounded or Infeasible . . . .	24
Table 4.	Conditions for Potential Infeasibility or Redundancy in Equations Based on Bounds on Variables . . . . .	25

## **GAMSCHK USER DOCUMENTATION**

This document describes procedures designed to aid users who wish to examine empirical GAMS models for possible flaws. The conceptual basis for many of the routines herein is supplied in McCarl and Spreen, and McCarl et.al.

This package of routines is designed for use on any GAMS platform, but for now is implemented on the HP, PC, DEC Alpha, IBM RS6000 and SUN workstations. The function of the specific components of GAMSCHK are to:

- (a) List coefficients for user selected equations and/or variables using the DISPLAYCR procedure.
- (b) List the characteristics of selected groups of variables and/or equations using MATCHIT.
- (c) List the characteristics of equation and variable blocks using BLOCKLIST.
- (d) Examine a GAMS model to see whether any variables and equations contain specification errors using ANALYSIS.
- (e) Generate schematics depicting the characteristics of coefficients by variable and equation blocks using BLOCKPIC.
- (f) Generate a schematic for small GAMS models or portions of larger models depicting the location of coefficients by sign and magnitude using PICTURE.
- (g) Reconstruct the reduced cost of variables and the activity within equations after a model solution using POSTOPT.
- (h) Help resolving problems with unbounded or infeasible models using NONOPT and ADVISORY.

### **General Notes on Package Usage**

GAMSCHK must replace a solver. This is done using a GAMS option statement of the form:

```
OPTION LP= GAMSCHK;  
      or  
OPTION NLP=GAMSCHK;  
      or  
OPTION MIP=GAMSCHK;
```

which replaces either the NLP, LP, or MIP solver with GAMSCHK.<sup>1</sup> In turn, the user will invoke the solver using the statement:

```
SOLVE MODELNAME USING LP MINIMIZING OBJNAME;
```

where MODELNAME is the name used in the GAMS MODEL statement; OBJNAME is the objective function name for the model; and the type of solver that GAMSCHK has replaced which must also be able to solve this type of problem (LP, NLP, or MIP) is identified.

The following are examples of GAMS sequences which can be added to the GAMS file:

```
OPTION NLP=GAMSCHK;  
SOLVE TRANSPORT USING NLP MINIMIZING Z;  
or  
OPTION LP=GAMSCHK;  
SOLVE FEED USING LP MINIMIZING COST;  
or  
OPTION MIP=GAMSCHK;  
SOLVE RESOURCE USING MIP MAXIMIZING PROFIT;
```

### **Selecting a Procedure and Providing Input -- the \*.GCK File**

GAMSCHK requires that the user indicate which procedures are to be employed. This is specified through the use of the \*.GCK file where the \* refers to the filename from the GAMS execution instruction<sup>2</sup>. The general form of that file is:

```
FIRST PROCEDURE NAME  
ITEM SELECTION INPUT
```

```
SECOND PROCEDURE NAME  
ITEM SELECTION INPUT
```

Spaces and capitalization are ignored in this input. For example, a \*.GCK file could look like

```
DISPLAYCR
```

---

<sup>1</sup> In all cases, users will be able to replace the LP solver. Replacement of the other solvers depends on the solver licenses owned by the user.

<sup>2</sup> Thus, if the GAMS instructions are in the file called MYMODEL, and GAMS is invoked using the DOS command GAMS MYMODEL, then the GCK file would be called MYMODEL.GCK. If GAMS instructions are on the filename with a period in it then the name up to the period will be used, i.e., the GCK file associated with MYMODEL.IT would be MYMODEL.GCK

```

variables
    SELL(*,*,FANCY)
    maketable
Invariables
    transport(plant2,*,fancy)
Equations
    objT
    notthere
inequations
    resourceq(plant1)

```

## PICTURE

The first procedure name in this case is DISPLAYCR and the following 10 lines indicate the items to be selected. Then, we also request PICTURE. Selection entries are treated using several assumptions. In particular:

- 1) If the \*.GCK file is empty then it is assumed that the BLOCKPIC procedure is selected.
- 2) Spaces maybe freely used in the GCK input file.
- 3) Upper, lower, or mixed case input is accepted.
- 4) GAMSCHK recognizes certain words. These words are listed in Appendix A and cannot be used as variable or equation names.

## The \*.GCK file: General Notes on Item Selection

Some of the procedures permit selection of variables, equations or functions. Specifically, the DISPLAYCR, PICTURE, POSTOPT, and MATCHIT procedures accept input identifying the variables and equations to be utilized. Also NONOPT accepts limited input controlling its function. General observations about the selection requests are

- 1) Variables can be chosen by entering the word *VARIABLE* or *VARIABLES* possibly with a modifier, followed by variable selection statements.
- 2) Variables can also be selected using the *INEQUATION* or *INEQUATIONS* syntax followed by names of equations. Use of this syntax results in selection of variables with coefficients in the named equations.
- 3) Equations are selected by entering the keyword, *EQUATION* or *EQUATIONS* possibly with a modifier, followed by equation selection statements.
- 4) Equations can also be selected using the *INVARIABLE* or *INVARIABLES* syntax followed by names of variables. Use of this syntax results in selection of equations in which the named variables have coefficients.

- 5) Certain item selection modifier keywords can be used depending on procedure. The *INTERSECT* keyword works with procedures *DISPLAYCR* and *POSTOPT*. The *INEQUATION* and *INVARIABLE* keywords work with procedures *DISPLAYCR*, *PICTURE* and *POSTOPT*. *LISTEQUATION* and *LISTVARIABLE* keywords work with the *MATCHIT* procedure. *INSOLUTION*, *NOTINSOLUTION*, *BINDING*, and *NOTBINDING* keywords work with *POSTOPT*. The keywords *VERBOSE* and *IDENTIFY* work with *NONOPT*.
- 6) If variable or equation names do not follow the keyword, then usually all variables or equations are assumed selected.

When variables or equations are to be selected after an item selection keyword, a number of input conventions apply. These conventions are:

- 1) If a variable or equation name is entered without any following parentheses, then all cases for that variable or equation are selected.
- 2) The selection entries identify specific elements from among the sets over which the variables and equations are defined. In specifying these elements one can use various wild card entries as discussed below or an element name.  
**Note GAMS set or subset names cannot be used. Set membership information is not available to the GAMSCHK routines.**
- 3) Wild cards can be used to select items. An "\*" will select any item. For example, "B\*" will select anything starting with a B. "A?B" will select anything beginning with A, ending with B with one intervening alpha numeric character.
- 4) When individual elements are specified, you need not enclose them in quotes (").
- 5) Quotes must be specified to include set item names with spaces, and special characters. In that case wild cards do not work and all input up to the next quote is simply copied.
- 6) When the selected item has more dimensions than specified, then all later dimensions are handled as if a wild card were specified. For example, when a variable X is defined with reference to 4 sets in the GAMS instructions, but only 3 parameters are specified in the GAMSCHK input, then the request is handled as if all elements of the 4th are desired.
- 7) When the selected item has less dimensions in GAMS than in the item

selection input, then all additional dimensions are ignored. Thus, when a variable X is defined with reference to 3 sets in GAMS, but 4 parameters are specified in the item selection file, then the 4th specification is ignored.

- 8) Multiple selection statements can appear on successive lines of the \*.GCK file. Output is ordered according to the way items are found in the GAMS file which is determined by the ordering of variables, equations, and set elements in the original GAMS input.
- 9) Error messages will be generated when an entry cannot be matched to a GAMS element.
- 10) Examples include

X(*,CLEVELAND)	which indicates that X will be selected for any element of the first set where the element in the second set equals CLEVELAND
X(SEATTLE)	when X is two dimensional selects all cases where the first set element is SEATTLE
X(SEATTLE,CHICAGO,Z)	when X is two dimensional selects the case where the first set element equals SEATTLE, and the second element equals CHICAGO. The third is ignored.
X	all X's will be selected
X(S*, C.O, Z)	when X is three dimensional selects where all X's with first element starting with S, second element beginning with C and ending with O and third element Z will be selected.
*	all variables or equations will be selected
{empty selection set}	all variables or equations will be selected

## Procedure Output

In all cases the output generated by the procedure will be written to the \*.LST file associated with the GAMS call. Thus, if the file is called MODEL with the \*.GCK file (MODEL.GCK), then all output will be on MODEL.LST.



## **Nonlinear Terms**

GAMS models examined with GAMSCHK may involve nonlinear terms. In such cases, GAMSCHK uses the value of the nonlinear term sent forth from GAMS which is an accurate marginal, not total value. GAMS develops this value based on the current level value of the variable. This will either be: a) the starting point selected by GAMS, if the model has not been solved, or b) the current solution value, if the model has been solved. The most accurate portrayals of the coefficients will be generated after the model has been solved through a GAMS SOLVE command before invoking GAMSCHK. Some cases may require a solution and/or the specification of a good starting point before using GAMSCHK. Also, nonlinear terms potentially cause misleading coefficients as those values are local marginal, not global, values determined by the current levels of the variables. Nonlinear terms are marked with \*\*\* in the DISPLAYCR, POSTOPT, and NONOPT output.

## **Entering Comments in the \*.GCK File**

The \*.GCK file has been programmed so that users can enter comments. These comments can take one of two forms. Comments that begin with a hash mark are copied to the output when the program runs. Comments which begin with a question mark are simply overlooked. Thus, one can temporarily comment GAMSCHK selection statements making them inactive by putting in question marks. If multiple procedures are being run or if some sort of output is decided to screen in the computer output then the hash marks can be entered.

## **Controlling Page Width in the \*.GCK File**

When running multiple procedures, in particular the pictures with other procedures, it is often desirable to have some procedures run with wide page widths, but the rest with a narrower page width. The GCK file provides the option to narrow the page width using a PW= command. In particular, what one can do is run GAMS with a large page width, i.e. run GAMS BLOCK pw=200, then insert in the GCK file instructions which narrow that page width for selected procedures. Users should note that the page width can never be made any wider than the default page width when running with GAMS. Information in excess of the page width will be ignored. Thus, if the model is run under the default status which has a page width of 75 characters then GAMSCHK will reduce the page width down to the maximum page width allowed. Consequently, the pw= command can only be used to narrow the page width from the default page width, not increase it.

## **Running Multiple Procedures**

GAMSCHK can run multiple procedures during one job. This is done by simply stacking the sequence of the commands in the .GCK file.

## Use of the Procedures

The following section describes the procedures available in GAMSCHK and their input requirements.

### DISPLAYCR

**Brief Purpose:** DISPLAYCR displays all coefficients from the empirical model for a set of user selected equations and variables. All nonzero coefficients under each selected variable or in each selected equation are displayed with the associated variable or equation name and coefficient value. The selection entries may refer to all terms in equations /under variables or only those coefficients at the intersection of the selected variables and equations.

**Usage Notes:** This option mirrors the GAMS LIMCOL and LIMROW options, but allows the user to select the specific items to be displayed. Partial displays within a variable or equation are also allowed using *INTERSECT*. Use of *VARIABLE* and *EQUATION* keywords followed by selection statements allows one to select variables and equations. Use of the *INVARIABLE* command allows users to select the equations which are associated with a particular variable. For example, if one is having trouble with a particular variable and wants to look at competition in the equations in which it appears, then selecting the variable under the *INVARIABLE* command will display the complete contents of all the equations in which the selected variables have coefficients. Similarly, the *INEQUATION* command will display the complete contents of all variables which fall in a particular equation. Nonlinear terms are marked with \*\*\*.

When the keyword *INTERSECT* is found then only the coefficients at the intersection of the specified equations and variables are selected. Use of *INTERSECT* with the *INVARIABLE* syntax results in the named variables and the equations in which they fall being selected. Similarly, use of *INTERSECT* with the *INEQUATION* syntax results in selection of the named equations and the variables which fall in those equations.

Note that when GAMS internal scaling features are employed

the default option is that the scaled output is displayed. This can be altered using the *DESCALE* feature of the solver options file.

Input File :

The keyword *DISPLAYCR* is entered followed by optional lines of item selection input identifying the variables and equations to be displayed. This file can contain the keywords *VARIABLE*, *INVARIABLE*, *EQUATION*, and *INEQUATION*, with each followed by a specification of the items to be selected using the procedure input specification conventions that were described above. The keyword *INTERSECT* can also be used. Several special cases are relevant:

- a) If none of the above keywords are found after *DISPLAYCR* and another procedure name does not follow, then the input is assumed to identify variables.
- b) If input is found but the *VARIABLE* or *INEQUATION* keyword cannot be found then no variables are assumed selected.
- c) If the *VARIABLE* keyword is entered, but is followed by the end of file or an Appendix A reserved word and *INEQUATION* does not appear, then all variables are assumed selected.
- d) If the *EQUATION* or *INVARIABLE* keyword cannot be found, then no equations are assumed selected.
- e) If the *EQUATION* keyword is entered, but is followed by the end of the file or a reserved word and the *INVARIABLE* command does not occur, then all equations are assumed selected.
- f) The keyword *INVARIABLE* is allowed. It should be followed by variable selection statements. In turn, *DISPLAYCR* selects all equations which have nonzero entries under the *INVARIABLE* selections.
- g) The keyword *INEQUATION* may be used. It should be followed by equation selection statements. In turn, *DISPLAYCR* selects all variables which have nonzero entries in the *INEQUATION* selections.

- h) The keyword *INTERSECT* causes only coefficients at the intersection of the specified equations and variables to be displayed. This occurs for all specifications in this run of DISPLAYCR. One should use DISPLAYCR again if some intersecting and some non-intersecting displays are desired.
- i) When *INTERSECT* appears along with *INVARIABLE*, the named variable is selected along with all the equations in which it falls. Similarly, when *INTERSECT* and *INEQUATION* appear then all the named equations and the variables appearing in them are selected.

## MATCHIT

**Brief Purpose:** MATCHIT retrieves the names and characteristics of selected variables and equations. The characteristics reported tell whether the items are nonlinear as well as reporting scaling characteristics and counts of the coefficients. MATCHIT will summarize the items which match a request or list all the items individually.

**Usage Notes:** The input to MATCHIT can include the keywords *VARIABLE* and *EQUATION* along with those keywords with the prefix *LIST* attached. When the *LIST* prefix is not used, the procedure summarizes the characteristics of all items which match the item requests counting the number of matching items, the number of those items which are nonlinear, the total coefficients under or in those items, the number of positive, negative, and nonlinear coefficients that fall under or in those items. This does not list the names of the individual items which match. If the *LIST* prefix is used (entering *LISTVARIABLE* or *LISTEQUATION*) then the individual matching items are printed in the order in which they are encountered. For each matching item the information tells whether it is nonlinear, how many total coefficients it has, the count of positive, negative, and nonlinear coefficients falling under it, and the minimum and maximum absolute values of coefficients under it (excluding the objective function coefficient).  
Note that when GAMS internal scaling features are employed then by default scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File : This file contains the keyword *MATCHIT*, followed by optional item selection input data. The optional input identifies the variables and equations to be displayed. This input can contain the keywords *VARIABLE* or *LISTVARIABLE* followed by a specification of the variables to be selected using the procedure input specification conventions that were described above. This can be followed by the keyword *EQUATION* or *LISTEQUATION* and the specified entries.

Several special cases are relevant:

- a) If the procedure name is not followed by any selection input, then a count of all variables and equations appears.
- b) If the input is found, but the input does not begin with *VARIABLE*, *EQUATION*, *LISTVARIABLE*, or *LISTEQUATION* keywords, then the input is assumed to contain variable names.
- c) If the *VARIABLE* keyword is entered, but is not followed by variable selection statements, and *LISTVARIABLE* does not appear, then all variables are assumed selected.
- d) If the *EQUATION* or *LISTEQUATION* keyword cannot be found, then no equations are assumed selected.
- e) If the *EQUATION* keyword is entered, but is not followed by equation selection statements or a *LISTEQUATION* entry, then all equations are assumed selected.
- f) The keyword *LISTVARIABLE* is allowed. It should be followed by variable selection statements. In turn, *MATCHIT* lists all variables which fall under the request.
- g) The keyword *LISTEQUATION* may also be used. It should be followed by equation selection statements. In turn, *MATCHIT* lists all equations which fall under the request.

- h) If none of the above keywords are found, the input is assumed to identify variables.

## **ANALYSIS**

**Brief Purpose:** Analyzes the structure of all variables and equations. Information is given on errors involving obvious model misspecifications causing redundancy, zero variable values, infeasibility, unboundedness, or obvious constraint relaxations in linear programs. The checks are those identified in Tables 1, 2 and 3.

**Usage Notes:** The analysis tests given in Tables 1 and 2 are utilized to determine if individual variables or equations in the model possess obvious specification errors. One test, for example, considers whether or not in a maximization problem a variable appears which has a positive return in the objective function, but no coefficients in the constraints indicating an obviously unbounded model. Similarly, information is provided on whether certain equations can never be satisfied. For example, tests examine whether an equality equation appears with a negative right hand side and all positives on the left hand side. Also tests see whether the bounds on variables preclude equation satisfaction or make equations redundant (Table 3). In ANALYSIS these tests are applied to each and every variable and equation. The BLOCKPIC and BLOCKLIST routines utilize the tests on a block by block basis. Thus, the messages will be triggered only if every variable or equation in that block has the same problem. Also interactions between variables and equations are not checked so ANALYSIS only finds flaws contained in individual variables/equations.

**Input File:** The keyword ANALYSIS is all that is accepted.

## **BLOCKLIST**

**Brief Purpose :** The BLOCKLIST procedure displays the number and characteristics of the items in each GAMS variable and equation block.

**Usage Notes:** The characteristic information gives:

- 1) The variable sign restriction or equation inequality type.
- 2) The number of variables or equations in this block;

- 3) The number of variables or equations with at least one nonlinear term in this block.
- 4) The number of positive coefficients under the variables or in the equations.
- 5) The number of negative coefficients under the variables or in the equations.
- 6) The number of nonlinear coefficients under the variables or in the equations.
- 7) The largest coefficient in absolute value in this block;
- 8) The smallest coefficient in absolute value in this block. Analysis tests are also performed as discussed under the ANALYSIS procedure.

Note that when GAMS internal scaling features are employed, the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File: No input other than the procedure name is needed.

## **BLOCKPIC**

Brief Purpose: Generates model schematics and scaling information. The schematics depict coefficient signs, total and average number of coefficients within each GAMS equation and variable block.

Usage Notes: These schematics are designed to aid users in identifying flaws in coefficient placement and sign. The summary information on problem scaling characteristics is designed to help users in scaling data. The scaling information is usually reported after any GAMS scaling (using the `variablename.scale` and `equationname.scale` features) but before solver scaling. (The user can change whether descaling is done - see the options file). Analysis tests are done using the procedures in Tables 1 and 2.

Note that when GAMS internal scaling features are employed the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver

options file.

Input File :           The keyword BLOCKPIC is all that is recognized.

## PICTURE

**Brief Purpose:** Generates a schematic depicting the location, sign and magnitude of coefficients for selected variables and equations. Users can use this schematic to help identify flaws in coefficient placement, magnitude, or sign. Reports are also generated on the number of individual elements in the pictured portions of each variable and equation.

**Usage Notes:**           This output can be quite large, so PICTURE should only be used for small models or model components.

Note that when GAMS internal scaling features are employed, the default option is that the scaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

**Input File:**           Optional input instructions may appear after the PICTURE keyword. This input selects the variables and equations to be included. Only coefficients at the intersection of the selected variables and equations are portrayed. The selected item in the .GCK file can contain the keywords *VARIABLE*, or *INVARIABLE* followed by a specification of the selected variables using the procedure input specification conventions above. This can be followed by the keywords *EQUATION* or *INEQUATION* and the specified entries. Several special cases are also relevant:

- a)     If the *VARIABLE* or *INEQUATION* keywords cannot be found, then all variables are assumed selected.
- b)     If the *EQUATION* or *INVARIABLE* keywords cannot be found, then all equations are assumed to selected.
- c)     If the none of the *VARIABLE*, *INVARIABLE*, *EQUATION*, or *INEQUATION* keywords are found, everything is pictured and all other input is ignored.
- d)     When the *INVARIABLE* keyword is used, then all equations in which those variables have coefficients are selected along with the named variables.



- e) When the *INEQUATION* keyword is used, then all variables which have coefficients in the named equations are selected along with the named equations.

## POSTOPT

Brief Purpose:

Does post optimality computations. In that capacity POSTOPT either:

- a) Reconstructs the reduced cost of variables after a GAMS model solution. Modelers can use this information to discover why certain variables are nonbasic or why certain shadow prices take on particular values, or
- b) Reconstructs the usage and supply across an equation after a GAMS model solution. Modelers can use this information to discover why certain variables or slacks take on particular values, as well as to find out where items within equations are produced and/or used.

Usage Notes:

POSTOPT uses essentially the same input conventions as does DISPLAYCR. Thus, the usage notes in that selection are also relevant here. In addition:

- 1) POSTOPT requires a solution has been obtained GAMSCHK will automatically cause a solver to be invoked unless suppressed by the options file;
- 2) Nonlinear terms may not be accurate in the row sums as their marginal value not their total value is used but GAMS will have adjusted the right-hand sides for their presence; and
- 3) Attention can be restricted to only certain types of variables or equations. Variables that are *INSOLUTION* (Nonzero or with Zero marginals), *NOTINSOLUTION* (zero with a nonzero marginal) can be requested, *BINDING* or *NONBINDING* equations can be focussed on.

Note that when GAMS internal scaling features are employed, the default option is that the unscaled output is displayed. This can be altered using the DESCALE feature of the solver options file.

Input File : An optional input file is read in, indicating the specific variables desired using the conventions explained under DISPLAYCR above. In addition:

- 1) One can enter *INSOLUTION* to restrict attention to variables which are nonzero or have zero marginals.
- 2) One can enter *NOTINSOLUTION* to restrict attention to zero variables.
- 3) The above entries restrict alteration in all *VARIABLE* or *INEQUATION* selection statements in a POSTOPT run.
- 4) One can enter *BINDING* to only consider equations with zero slack. Similarly, *NONBINDING* considers equations with nonzero slack.
- 5) The above equation specifications restrict all sections by all *EQUATION* or *INVARIABLES* items in a POSTOPT run.

## ADVISORY

Brief Purpose: To identify variables which could be unbounded or equations and variable bounds which could cause a model to be infeasible.

Usage Notes: The ADVISORY procedure causes a presolution report on the set of all: a) variables which could be unbounded and/or b) equations and variable bounds which could cause infeasibility. The tests used are summarized in Table 3. This procedure identifies all variables which would need to be bounded as well as all constraints which need artificial variables if one wishes to diagnose problems in a model. The same output is also generated by NONOPT but the ADVISORY version does not require a solution.

Input file: Just the word ADVISORY

## NONOPT

Brief Purpose: To help diagnose unbounded and infeasible models.

Usage Notes: The NONOPT procedure can be used in either an informative mode or with models which terminate as unbounded or infeasible. NONOPT will look through an optimal model reporting all variables which may be potentially unbounded or infeasible and all equations which may be infeasible using the checks explained under the ADVISORY section. Also in an unbounded model NONOPT can report the names of unbounded or infeasible variables or equations as well as either budgeting or row summing them. NONOPT runs after a solution and causes a solve to occur.

Input File: NONOPT may be followed by optional keywords *IDENTIFY* or *VERBOSE*. The *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. Only the last encountered of the *VERBOSE* or *IDENTIFY* keywords will be obeyed. The details on these options are as follows:

- 1) If the *IDENTIFY* keyword is used, then the rules in Table 3 are applied to the model. Identify also anticipates that large upper bounds and/or artificial variables may be present. In an optimal condition all variable and equation levels that have exponents greater than the user supplied level filter in the options file (or 6 by default) are identified as items which could be involved with an unbounded model. Similarly, all variables or equations with marginals greater in exponent than the user supplied marginal exponent filter will be identified as items potentially involved with an infeasible model.
- 2) When the *VERBOSE* keyword is read then all variables and equations which are listed as nonoptimal or infeasible are treated using the budgeting and row summing aspects of POSTOPT.
- 3) When no keyword is found and the model solution is not

optimal then the nonoptimal equations, infeasible equations and/or nonoptimal variables are automatically listed.

### Options File

GAMSCHK accepts an option file controlling solver choice (when needed); descaling; and size of the nonoptimal filters; the number of variable and column blocks selection entries allowed. The file is called GAMSCHK.OPT.

### Solver Choice Options

GAMSCHK calls for the solution of the problem when the POSTOPT or NONOPT procedures are used. In doing this, GAMSCHK internally selects the default GAMS solver for a problem class. Users may override this choice using the solver options file. Users may also force or suppress the solution process.

There are 9 solver related keywords allowed in the options file. These are as follows:

<i>OPTION</i>	<i>Purpose</i>
<i>LP</i>	Gives name of solver for LP problems
<i>NLP</i>	Gives name of solver for NLP problems
<i>MIP</i>	Gives name of solver for MIP problems
<i>DNLP</i>	Gives name of solver for DNLP problems
<i>SOLVERNAME</i>	Gives name of solver to be used regardless of problem type
<i>NOSOLVE</i>	Suppresses solution of the problem
<i>SOLVE</i>	Forces solution of the problem
<i>DESCALE</i>	Controls treatment of scaling
<i>OPTFILE</i>	Solver options file number

In the first five cases, the option name is followed by the name of one of the licensed solvers. If the options file is empty, then the default solver will be used. If a solver name is given, then that solver will be used provided it matches the name of a solver GAMS recognizes.

### When Should I Use *SOLVE* or *NOSOLVE*

Ordinarily GAMSCHK will cause a solver to be used if either the POSTOPT or the NONOPT options are used. However, users can force solutions under other cases or suppress solutions if desired.

One should only force a solution (using the *SOLVE* option) when one wishes to use the solution information after GAMSCHK is done either to examine the solution output or do post optimality calculations. Forcing a solution will not cause GAMSCHK to have improved

representations of nonlinear terms. That will only occur when a *SOLVE* statement is executed before the *SOLVE* statement involving GAMSCHK.

### Control of Number of Variable and Row Selections Allowed

The GAMSCHK program uses an upper estimate on the number of variable or equation blocks. In rare circumstances users may wish to override this choice. The options for this are

<i>OPTION</i>	<i>Purpose</i>
<i>VARBLOCK</i>	Maximum number of variable blocks allowed
<i>EQUBLOCK</i>	Maximum number of equation blocks allowed

These options are followed by a number, but should not be routinely used.

### Scaling

GAMS users may be utilizing internal features which involve scaling through the Modelname.SCALEOPT=1, VariableName.SCALE, and EquationName.SCALE options. GAMSCHK can work with these options to create output which reflects scaled, unscaled or partially unscaled output. In particular, the command DESCALE can be entered with one of three options: *NEVER*, *ALL*, or *PART*. If you enter *NEVER*, then none of the model output will be descaled. If you enter *ALL*, then all of the model output will be descaled. The third option is to use *PART*. In that case the NONOPT and POSTOPT output will be descaled whereas scaled information will be displayed for PICTURE, BLOCKPIC, BLOCKLIST, MATCHIT and DISPLAYCR. The *PART* option allows investigation of scaling. If you do not enter a *DESCALE* option then all information will be reported as if the *PART* option was chosen.

### NONOPT Filters

The NONOPT model in “*IDENTIFY*” mode checks through a model solution to identify large marginals and/or large variable values. The limits on these checks are provided by two options

<i>OPTION</i>	<i>Purpose</i>
<i>LEVELFILT</i>	Numerical value of exponent on “unbounded levels”
<i>MARGFILT</i>	Numerical value of exponent on “infeasible marginals”

These options provide upper bounds on the exponents of the absolute values for the levels and marginals. They are followed by an integer which gives the exponent. Thus, entries like

<i>LEVELFILT</i>	7
<i>MARGFILT</i>	7

will cause the reporting of all marginals and levels which are greater in absolute value than  $10^7$

### Example Options File

The GAMSCHK option file is called GAMSCHK.OPT. An example of a file could look like the following 6 lines

LP	OSL
MIP	LAMPS
VARBLOCK	50
SOLVE	
DESCALE	PART
LEVELFILT	4

### Solver Options File

One other important aspect regarding the options file involves the use of a problem solver options file when a solver such as MINOS5, OSL, LAMPS etc. is also being used. As seen above the GAMSCHK.OPT does not recognize option commands such as those which would be submitted to the programming model solvers - OSL for example. In all cases GAMSCHK will cause the default option file for the solver to be used when invoking the solver. Thus if MINOS5 and the options file is invoked is being used, MINOS5 options are controlled by the option file MINO5.OPT while GAMSCHK.OPT controls GAMSCHK operation.

Users can change the number of the solver options file being used by using the OPTFILE parameter in the options file. OPTFILE 2 would cause use of solver options file .OP2.

### Known Bugs

There are a few bugs that can cause GAMSCHK to report improper outputs or results. A list of the known bugs, their symptoms and a remedy is given below.

Symptom	Cause	Remedy
---------	-------	--------

Zero Shadow Prices in POSTOPT	Old GAMS version or no Prior Solve	1) Make sure the model was solved, 2) if it was, do not suppress solve in option file, or 3) update to most recent GAMS version
Decaling Does Not Work	Old Version of GAMS	Update
GAMS Blows up after GAMSCHK Runs	Old GAMS version	Ignore, *.LST file, results are fine, can be fixed by updating to the most recent version of GAMS
POSTOPT has error in budgets equal to twice objective function coefficient for nonlinear maximizations	Old GAMS MINOS version	Switch to a minimization formulation or update GAMS/MINOS
ROWSUM does not fully account for the value of nonlinear terms in POSTOPT	Value of nonlinear terms sent from GAMS are only a marginal value	None planned. GAMSCHK would need reprogramming
Error message about size of VARBLOCK or EQNBLOCK	exceeded maximum number of blocks	Modify option file, enlarging or eliminating parameters
GAMSCHK won't run	Files are not properly installed	Recheck installation. If still doesn't work report to author
Zero shadow prices when using NOSOLVE	Old version of GAMS solvers or Shadow prices suppressed	Try changing GAMSCOMP.TXT lines 2 or 0 to 12 or 10, if that doesn't work update GAMS.

## References

- Brooke, A., D. Kendrick, and A. Meeraus. GAMS: A User's Guide. The Scientific Press, South San Francisco, CA, 1988.
- McCarl, B.A. "So Your GAMS Model Didn't Work Right: A Guide to Model Repair." Texas A&M University, College Station, TX, 1994.
- McCarl, B.A., and T.H. Spreen. "Applied Mathematical Programming Using Algebraic Systems." Draft Book, Department of Agricultural Economics, Texas A&M University, College Station, TX, 1996. On web page <http://agrinet.tamu.edu/mccarl>



Table 1. Conditions under which a modeler should be advised of potential difficulty for equations without nonlinear terms.

Type of Constraint	Count of coefficients under a variable of this type with a particular sign						Sign of RHS	Type of PS <sup>a/</sup>	Examples <sup>b/</sup>
	Nonnegative		Nonpositive		Unrestricted				
	+	-	+	-	+	-			
≤	≥ 0 <sup>c/</sup>	0	0	≥ 0	0	0	0	Zero Variables - Case 1	∑ x ≤ 0 <sup>d/</sup> , -∑ y ≤ 0 , ∑ x - ∑ y ≤ 0
	≥ 0	0	0	≥ 0	0	0	-	Infeasible -Case 2	∑ x ≤ - k , -∑ y ≤ - k , ∑ x - ∑ y ≤ - k
	0	≥ 0	≥ 0	0	0	0	+ or 0	Redundant -Case 3	-∑ x ≤ + k , ∑ y ≤ + k, -∑ x + ∑ y ≤ k
=	≥ 0	0	0	≥ 0	0	0	0	Zero Variables - Case 1	∑ x = 0 , -∑ y = 0, ∑ x - ∑ y = 0
	0	≥ 0	≥ 0	0	0	0	0	Zero Variables - Case 1	-∑ x = 0 , ∑ y = 0, -∑ x + ∑ y = 0
	≥ 0	0	0	≥ 0	0	0	-	Infeasible -Case 2	∑ x - ∑ y = -k
	0	≥ 0	≥ 0	0	0	0	+	Infeasible -Case 2	-∑ x + ∑ y = k
	0	0	0	0	≥ 0 <sup>e/</sup>	≥ 0 <sup>e/</sup>	0	Zero Variable - Case 1	z=0 , -z=0
≥	0	≥ 0	≥ 0	0	0	0	0	Zero Variables - Case 1	-∑ x ≥ 0 , ∑ y ≥ 0 , -∑ x + ∑ y ≥ 0
	0	≥ 0	≥ 0	0	0	0	0 or +	Infeasible -Case 2	-∑ x ≥ k , ∑ y ≥ k , -∑ x + ∑ y ≥ k
	≥ 0	0	0	≥ 0	0	0	- or 0	Redundant -Case 3	∑ x ≥ - k , -∑ y ≥ - k , ∑ x - ∑ y ≥ - k

- <sup>a/</sup> The PS cases indicate, because the variables in this equation follow this pattern, that:
1. The variables appearing with nonzeros in this equation are forced to equal zero.
  2. This equation can never be satisfied and is obviously infeasible.
  3. This equation is redundant. The nonnegativity conditions are a stronger restriction.

<sup>b/</sup> In the examples x denotes indexed non-negative variables, y indexed non-positive variables, and z a single unrestricted variable.

<sup>c/</sup> Here and in the cases below at least one nonzero must occur

<sup>d/</sup> These entries give examples of the problem covered by each warning. Namely, in the first case examining only the nonnegative variables suppose all those variables have signs  $\geq 0$  but the right-hand-side is zero. Thus, we have  $X \geq 0$  and  $X \leq 0$  which implies  $X = 0$ . A warning is generated in that case.

<sup>e/</sup> Only one coefficient is allowed.

Table 2. Conditions under which a modeler should be warned about variables in a maximization problem.

Type of Variable	Objective function coefficient sign	Number of $a_{ij}$ 's of a sign in						PS <sup>a/</sup>	Examples
		$\geq$ rows		$=$ rows		in $\leq$ rows			
		+	-	+	-	+	-		
Nonnegative	+	$\geq 0$	0	0	0	0	$\geq 0$	Unbounded Variable case 1	$\max x^{b/}$ , $x + DQ \geq a$ $-x + EQ \leq b$
	-	0	$\geq 0$	0	0	$\geq 0$	0	Zero optimal solution case 2	$\max -x$ $-x + DQ \geq a$ $x + EQ \leq b$
	0	$\geq 0$	0	0	0	0	$\geq 0$	Variable Relaxes constraint case 3	$\max 0x$ $x + DQ \geq a$ $-x + DQ \leq b$
	0	$\geq 0$	0	$\geq 0^c$	$\geq 0^c$	0	$\geq 0$	Variable Relaxes constraint case 4	$\max 0x$ $x + DQ \geq a$ $x + FQ = g$ $-x + EQ \leq b$
Nonpositive	-	0	$\geq 0$	0	0	$\geq 0$	0	Unbounded Variable case 1	$\max -y$ $-y + DQ \geq a$ $y + EQ \leq b$
	+	$\geq 0$	0	0	0	0	$\geq 0$	Zero optimal solution case 2	$\max y^{b/}$ , $y + DQ \geq a$ $-y + EQ \leq b$
	0	0	$\geq 0$	0	0	$\geq 0$	0	Variable Relaxes constraint case 3	$\max 0x$ $-y + DQ \geq a$ $y + EQ \leq b$
	0	0	$\geq 0$	$\geq 0^c$	$\geq 0^c$	$\geq 0$	0	Variable Relaxes constraint case 4	$\max 0x$ $-y + DQ \geq a$ $y + FQ = g$ $y + EQ \leq b$
Unrestricted	+/-	0	0	0	0	0	0	Unbounded Variable case 1	$\max \pm z$

<sup>a/</sup> PS cases are: The variables which satisfy this condition are:

- 1) Unbounded as they contribute to the objective function while satisfying the constraints.
- 2) Obviously zero since they consume constraint resources and have a cost in the objective function.
- 3) Warning this variable relaxes all constraints in which it appears
- 4) Warning this variable relaxes all the equality constraints in which it appears in one direction

<sup>b/</sup> Here  $x(y)$  has a positive objective term and can be increased without ever violating any constraints so  $x(y)$  is unbounded.

<sup>c/</sup> Only one coefficient can be present in the equality rows

Table 3. Conditions When Model Elements Could be Unbounded or Infeasible

Conditions for Potential Unbounded Variables -- Presence of Bounds

Types of Variables	Sign of Objective in Max Problem	Upper	Lower
$\geq 0^a$	+	None	--- <sup>c</sup>
$\leq 0$	-	---	None
Unrestricted	+	None	---
Unrestricted	---	---	None

Conditions for Potential Infeasibility Caused by Bounds on Variables

Existence of Bounds		
Types of Variables	Lower	Upper
$\geq {}^b0$	+	---
$\leq 0$	---	---
Unrestricted	+	---
Unrestricted	---	---

Conditions for Potential Infeasibility in Equations

Types of Equations	RHS
$\leq {}^d$	-
$\geq$	+
=	+ or -

- <sup>a</sup> If a non negative variable has a positive objective function coefficient without an upper bound, then the variable could be unbounded.
- <sup>b</sup> If a nonnegative variable has a positive lower bound then it could cause infeasibility.
- <sup>c</sup> Any reasonable value can exist for this item
- <sup>d</sup> When a less than or equal equation is present it may not be able to be satisfied if it has a negative RHS.

Table 4. Conditions for Potential Infeasibility or Redundancy in Equations Based on Bounds on Variables

	TYPE OF CONSTRAINT		PS
	$\leq b$	$\geq b$	
SUM OF THE SMALLEST VALUE <sup>a</sup>	$> b$	---	INFEASIBLE
	---	$> b$	REDUNDANT
SUM OF THE LARGEST VALUE <sup>b</sup>	---	$< b$	INFEASIBLE
	$< b$	---	REDUNDANT

Note:

a. Suppose  $X_j$  is bounded as follows,  $LB_j$  (lower bound)  $\leq X_j \leq UB_j$  (upper bound), and we have the sum which is either  $>b$  or  $<b$ , then

$S=$  will be the smallest value which could happen in that sum. If the constraint is  $\leq b$ , then if  $S > b$ , we know that this constraint will never be satisfied. In the constraint is  $\geq b$ , then if  $S > b$ , we know that this constraint will not limit any possible  $X$  value. Hence, it is redundant.

b. Suppose  $X_j$  is bounded as follows,  $LB_j$  (lower bound)  $\leq X_j \leq UB_j$  (upper bound), and we have the sum which is either  $>b$  or  $<b$ , then

$L=$  will be the largest value which could happen in that sum. If the constraint is  $\leq b$ , if  $L < b$ , we know that this constraint will not limit any possible  $X$  value. Hence, it is redundant. In the constraint is  $\geq b$ , then if  $L < b$ , we know that this constraint will never be satisfied.

c. Thanks to Paul Preckel for bringing these tests to the authors' attention.

## Appendix A: Reserved Names

VARIABLE  
VARIABLES  
EQUATION  
EQUATIONS  
INVARIABLE  
INVARIABLES  
INEQUATION  
INEQUATIONS  
LISTVARIABLE  
LISTVARIABLES  
LISTEQUATION  
LISTEQUATIONS  
POSTOPT  
DISPLAYCR  
PICTURE  
BLOCKPIC  
ANALYSIS  
MATCHIT  
BLOCKLIST  
NONOPT  
INSOLUTION  
NOTINSOLUTION  
NONINSOLUTON  
VERBOSE  
ADVISORY  
BINDING  
NONBINDING  
NOTBINDING  
INTERSECT  
IDENTIFY  
PW=

## Appendix B: GAMSCHK One Page Summary

**Invoking GAMSCHK**    **OPTION LP=GAMSCHK**

**Keywords allowed in GCK file**

Keyword	Allowed SubKEYWORDS	Brief Description
DISPLAYCR	VARIABLE* INVARIABLE* EQUATION* INEQUATION* INTERSECT <sup>++</sup>	Displays coefficients of selected variables and equations Indicates variable selections follow Indicates equations are wanted in which selected variables fall Indicates equation selections follow Indicates variables are wanted that fall in selected equations Show coefficients which appear at intersections of selected var/eqn
MATCHIT	VARIABLE* LISTVARIABLE* EQUATION* LISTEQUATION*	List variable and equation names and summarize characteristics Summarizes all variables matching selection statements Lists each variable matching a selection statement Summarizes all equations matching a selection statement Lists each equation matching a selection statement
ANALYSIS		Checks for obvious structural defects
BLOCKLIST		Summarizes characteristics of variable and equation blocks
BLOCKPIC		Generates block level schematics
PICTURE	VARIABLE* INVARIABLE* EQUATION* INEQUATION*	Generates tableau schematics Indicates variable selections follow Indicates equations are wanted in which selected variables fall Indicates equation selections follow Indicates variables are wanted that fall in selected equations
POSTOPT	VARIABLE* INVARIABLE* EQUATION* INEQUATION* INTERSECT <sup>++</sup> NOTINSOLUTION <sup>++</sup> INSOLUTION <sup>++</sup> BINDING <sup>++</sup> NONBINDING <sup>++</sup>	Reconstructs reduced cost and equation activity Indicates variable selections follow Indicates equations are wanted in which selected variables fall Indicates equation selections follow Indicates variables are wanted that fall in selected equations Show coefficients which appear at intersections of selected var/eqn Only nonzero vars or those with zero reduced cost Only zero vars will be selected Only eqns with zero slack will be computed Only eqns with nonzero slack will be computed
ADVISORY		List potential infeasible and unbounded items
NONOPT	IDENTIFY VERBOSE	Lists potential or actual nonoptimal items Same as ADVISORY but after solution Does POSTOPT computations on nonoptimals

### Other Notes

- 1) Items marked above with an \* are followed by item selection statements.
- 2) Items marked with ++ modify the types of variables, equations and coefficients selected.
- 3) In item selection an \* is a wild card for multiple characters while a . is a wildcard for one character.
- 4) Spaces and capitalization don't matter in any of the input.
- 5) Options file controls scaling, solver choice, nonopt filters and maximum allowed selections.
- 6) Page width is controlled by a PW= keyword but cannot exceed GAMS page width.
- 7) Lines beginning with a ? or a # are treated as comments.