

Esta obra esta bajo una licencia reconocimiento-no comercial 2.5 Colombia de creativecommons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/2.5/co/> o envíe una carta a creative commons, 171second street, suite 30 San Francisco, California 94105, USA

GAMS Aplicado a las Ciencias económicas



Autores:

Juan Carlos Tarapuez Roa
Gloria Stella Barrera Ardila

Director Unidad Informática: Henry Martínez Sarmiento

Tutor Investigación: Alejandro Nieto Ramos

Coordinadores: Alejandro Nieto Ramos
Laura Vanessa Hernández
Juan Felipe Reyes Rodríguez

Coordinador Servicios Web: Miguel Ibáñez

**Analista de Infraestructura
y Comunicaciones:** Alejandro Bolívar

**Analista de Sistemas de
Información:** Mesías Anacona Obando

Coordinadora Inventarios: Sandra Yazmin Corrales

UNIVERSIDAD NACIONAL COLOMBIA
FACULTAD DE CIENCIAS ECONÓMICAS
UNIDAD DE INFORMÁTICA Y COMUNICACIONES
BOGOTÁ D.C.
2010

GAMS Aplicado a las Ciencias económicas

Director Unidad Informática: Henry Martínez Sarmiento

Tutor Investigación: Alejandro Nieto Ramos

Auxiliares de Investigación:

GLORIA STELLA	BARRERA ARDILA
DAVID FELIPE	BELTRAN GOMEZ
PEDRO ANDRES	BOHORQUEZ
IVAN ALBEIRO	CABEZAS MARTINEZ
SANDRA MILENA	CASTELLANOS PÁEZ
EDGAR ANDRES	GARCIA HERNANDEZ
CESAR LEONARDO	GARIBELLO OSPINA
FRANCISCO	GONZÁLEZ BUITRAGO
LILIANA CAROLINA	HERRERA PRIETO
LUIS CARLOS	MARTÍNEZ RUIZ
JEIMMY PAOLA	MUÑOZ SORACIPA
LINETH JOHANA	NIETO CHAVEZ
JAVIER ALEJANDRO	ORTIZ VARELA
CINDY LORENA	PABÓN GÓMEZ
JUAN DAVID	PÁEZ ALVAREZ
CAMILO ALEXANDRY	PEÑA TALERO
DIEGO ARMANDO	POVEDA ZAMORA
DANIEL FRANCISCO	ROJAS MARTÍN
JUAN CARLOS	TARAPUEZ ROA
CAMILO ALBERTO	ZAPATA MARTINEZ

Este trabajo es resultado del esfuerzo de todo el equipo perteneciente a la Unidad de Informática.

Se prohíbe la reproducción parcial o total de este documento, por cualquier tipo de método fotomecánico y/o electrónico, sin previa autorización de la Universidad Nacional de Colombia.

**UNIVERSIDAD NACIONAL COLOMBIA
FACULTAD DE CIENCIAS ECONÓMICAS
UNIDAD DE INFORMÁTICA Y COMUNICACIONES
BOGOTÁ D.C.
2010**

TABLA DE CONTENIDO

TABLA DE CONTENIDO	3
1. RESUMEN.....	4
2. ABSTRACT.....	4
3. INTRODUCCIÓN.....	5
4. GENERALIDADES.....	6
4.1. Instalación	6
4.2. Interfaz Gráfica	8
4.2.1. Barra de Menús	8
4.2.2. Área de trabajo	8
5. PROGRAMACIÓN	10
6. EJEMPLO	15
7. OUTPUTS Y ERRORES	17
8. GAMS EN INTERNET.....	18
9. LIBRERIAS	18
10. DESCRIPCION DE LOS PRINCIPALES SOLVERS Y ALGORITMOS DE RESOLUCIÓN DE GAMS.....	19
11. PAQUETES COMPLEMENTARIOS PARA CIENCIAS ECONÓMICAS (MPSGE/GAMS).....	29
11.1. Sintaxis con GAMS/MPSGE	31
12. CONCLUSIONES.....	36
13. BIBLIOGRAFÍA.....	36

1. RESUMEN

Este documento es una guía elaborada a manera de manual sobre el programa de simulación y modelación matemática GAMS (General Algebraic Modeling System) con aplicaciones dirigidas a los requerimientos de las ciencias económicas y de manera particular al equilibrio general computable. Hace parte de la construcción del curso libre de la UIFCE en GAMS y busca ser un elemento de apoyo fundamental para el mismo.

El manual da inicio presentando las generalidades de GAMS como el proceso de instalación y la estructura de su interfaz gráfica, continuando con la presentación de la estructura de programación y una descripción de los principales problemas de simulación y algoritmos de solución existentes, finalmente centra su atención en algunos complementos y utilidades culminando con el lenguaje MPSGE/GAMS.

2. ABSTRACT

This document is a tutorial about the simulation program and mathematical modeling GAMS (General Algebraic Modeling System) applications aimed at the requirements of economics and particularly the computable general equilibrium. Construction is part of the free course in GAMS UIFCE seeks to be an essential element of support for it.

The manual begins by presenting an overview of GAMS and the installation process and structure of its graphical interface, continuing with the presentation of the programming structure and a description of the main problems of simulation and existing solution algorithms finally focuses in some ins and utilities culminating with language MPSGE / GAMS.

3. INTRODUCCIÓN

Es un sistema de modelado de programación matemática que permite resolver problemas lineales, no lineales y de optimización. GAMS usa un lenguaje de modelización, en donde el usuario puede escribir en un editor la formulación del modelo matemático, y luego aplica un solver para resolver completamente el modelo, brindándole así un claro conocimiento de las condiciones de su modelo y de las consecuencias que tiene sobre él un cambio en el valor de cualquier variable.

El Sistema de Modelación Algebraica General o GAMS por sus siglas en inglés (General Algebraic Modeling System), fue diseñado especialmente para modelar sistemas lineales o no lineales, y para problemas de optimización. El programa responde muy bien para sistemas de gran tamaño, ahorrándole tiempo al usuario del programa en cosas repetitivas, ya que el programa se ocupa de hacer iteraciones y demás operaciones matemáticas que normalmente consumen mucho tiempo. El programa permite que el usuario pueda manipular a su gusto las variables del problema, creando así, en pocos segundos, un nuevo resultado con las nuevas variables y comparándola con el resultado inicial, mostrando los cambios del sistema; también en poco tiempo se puede cambiar el tipo de modelo, de uno lineal a otro no lineal o se pueden modificar las ecuaciones principales del problema; todo esto se logra de una manera fácil y rápida.

Con el uso de GAMS, el usuario se concentra más en la parte conceptual del modelo a desarrollar, ya que GAMS proporciona todas las herramientas cálculo para llevar a cabo cualquier modelo que el usuario programe. El lenguaje de GAMS es un lenguaje básico y altamente simplificado, haciendo que su manejo sea sencillo para personas que empiecen a usarlo. Además de su clara interfaz, GAMS ofrece un informe de resultados bastante explícito, donde muestra los resultados de las ecuaciones y las variables presentes en el modelo, y además de eso, nos muestra la comparación de los resultados obtenidos con los resultados anteriores, mostrando un cambio que finalmente se puede apreciar mejor en un informe elaborado por el usuario detalladamente.

Como el sistema fue desarrollado principalmente por economistas, posee muchas herramientas extras que llevan incorporados modelos económicos, permitiendo facilitar mucho más su uso y simplificar el tiempo de programación que en usuario le dedica. Sin embargo es un programa que tiene múltiples usos, no solo en el campo de la economía, sino que también en el campo de las matemáticas y en el de la ingeniería.

Como parte de su soporte teórico, GAMS cuenta con una completa base de datos incorporada, la cual tiene diversos modelos de optimización aplicados a muchos campos de estudio, y esto proporciona una gran ayuda al usuario ya que él se puede basar fácilmente en esos modelos para crear el suyo propio.

4. GENERALIDADES

4.1. Instalación

GAMS es un software de carácter propietario por lo que es necesario adquirir una licencia para hacer un uso total de este. Sin embargo los programadores de GAMS han dispuesto una distribución de demostración limitada por las siguientes condiciones:

Without a valid GAMS license the system will operate as a free demo system with these limitations:

Without a valid GAMS license the system will operate as a free demo system with these limitations:

1. **Model limits:**

- Number of constraints and variables: 300
- Number of nonzero elements: 2000 (of which 1000 nonlinear)
- Number of discrete variables: 50 (including semi continuous, semi integer and member of SOS-Sets)

2. **Global solver limits:**

- Number of constraints and variables: 10

The GAMS log will indicate that your system runs in demo mode:

```
GAMS Rev 232 Copyright (C) 1987-2009 GAMS Development. All rights reserved
Licensee: GAMS Development Corporation, Washington, DC G871201/0000CA-ANY
Free Demo, 202-342-0180, sales@GAMS.com, www.GAMS.com DC0000
```

```
--- Starting compilation
```

GAMS will terminate with a licensing error if you hit one of the limits above.

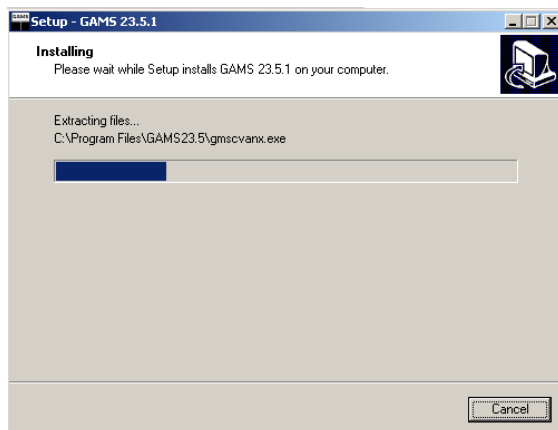
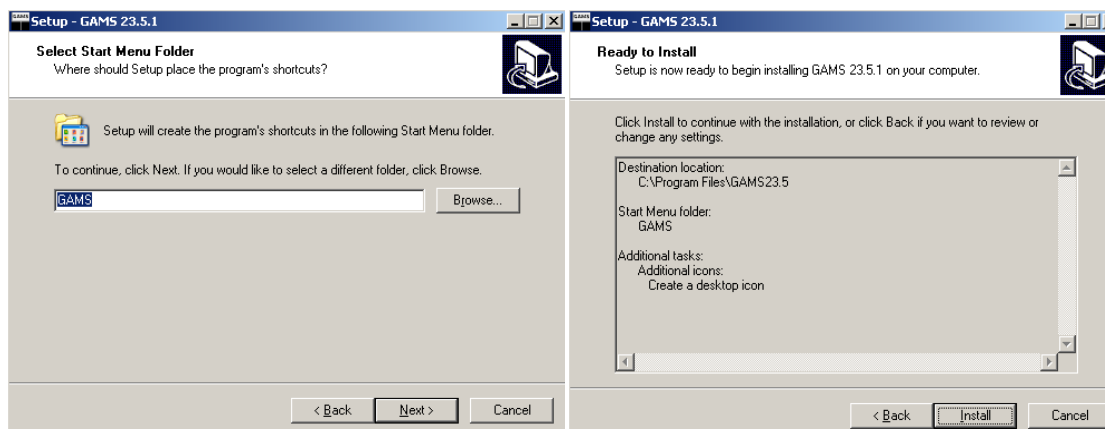
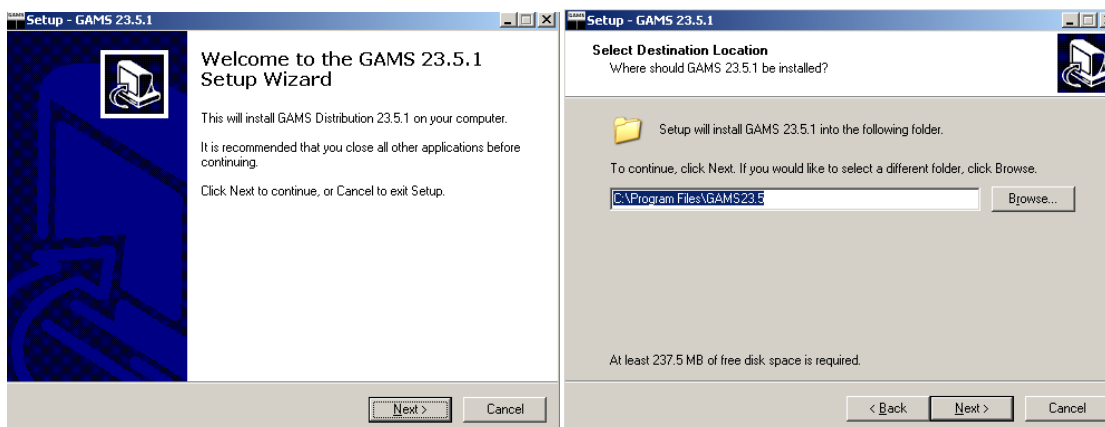
Tomado de <http://www.GAMS.com/download/>, Consultada el 25/08/2010.

La última versión¹ de GAMS puede ser adquirida desde la web oficial del programa en el link de descargas (<http://www.GAMS.com/download/>), allí se encuentran los archivos de instalación del software para un variado conjunto de sistemas operativos en sus principales arquitecturas.

A continuación se muestra el proceso de instalación de GAMS en Windows para una arquitectura de 32 bits:

¹ La web oficial de GAMS (<http://www.gams.com/>) provee también un conjunto de versiones anteriores que pueden ser descargadas.

Luego de descargar el archivo de la web de GAMS, se procede ejecutándolo, de tal forma que el asistente de instalación nos guíe.



Al terminar este paso, GAMS nos brindará la opción de licenciar el programa, lo que se logra con la selección de un archivo virtual en el que esta contenida la licencia (este archivo es proporcionado por la compañía que distribuye GAMS y requiere de un pago de licenciamiento), o de culminar con el proceso de instalación para trabajar con una versión de demostración. Sea cual sea la opción seleccionada, el proceso habrá finalizado y GAMS estará listo para ser utilizado.

4.2. Interfaz Gráfica

La interfaz gráfica de GAMS en Windows permite un sencillo manejo del programa.

4.2.1. Barra de Menús

La barra de menús esta constituida por los siguientes botones:

File: Muestra las opciones relacionadas con los archivos de trabajo de GAMS, abrir, guardar y compilar la información de los archivos de trabajo son algunas de las funciones mas importantes.

Edit: Este contiene las opciones relacionadas con la edición de los archivos que se están trabajando en GAMS.

Search: Compila las opciones de búsqueda y reemplazamiento de comandos y caracteres en la programación de los archivos de GAMS.

Windows: Contiene opciones para la organización de las subventanas del programa (Inputs, Reportes y ayuda).

Utilities: Cuenta con las opciones de grabado y edición de procedimientos en GAMS (son los macros de GAMS).

Model Libraries: GAMS tiene precargados unos ejemplos de programación y datos que la comunidad ha construido como parte complementaria y de ejemplificación.

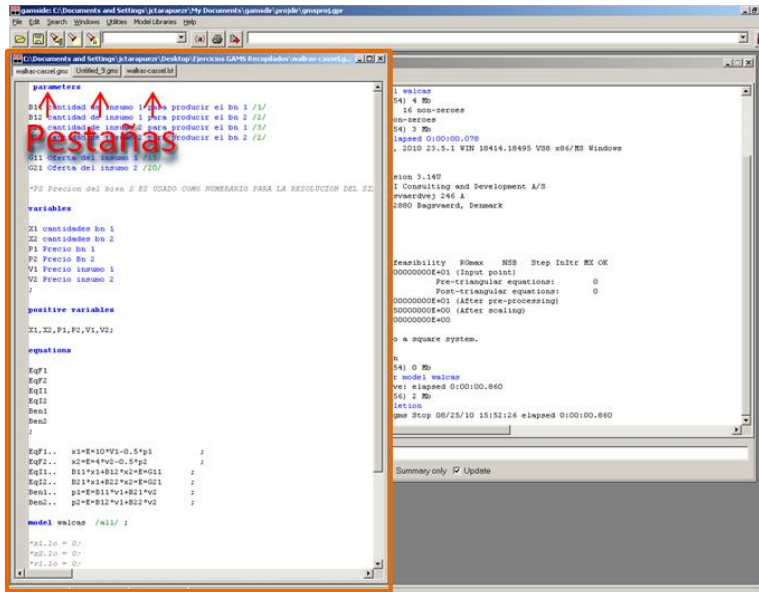
Help: Cuenta con diferentes opciones de ayuda, entre ellas un manual de referencia y un conjunto de tutoriales en PDF.



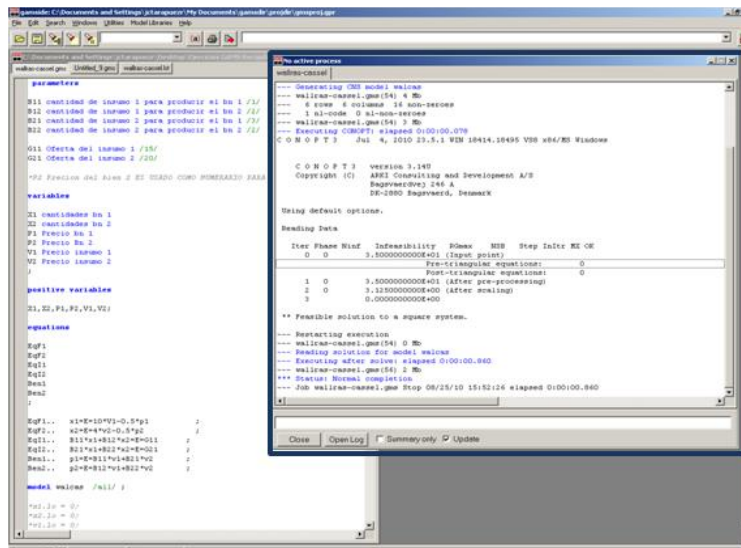
4.2.2. Área de trabajo

El área de trabajo se compone básicamente de dos grupos de ventanas, el primer grupo es el constituido por las ventanas de input y reportes de procesos, allí, se especifican todos los valores de programación que el usuario final requiera por cada simulación, es la puerta de acceso que el programador tiene con GAMS, pero también se especifican allí los reportes que GAMS hace tras cada compilación, estos reportes contienen por ejemplo los resultados de la simulación.

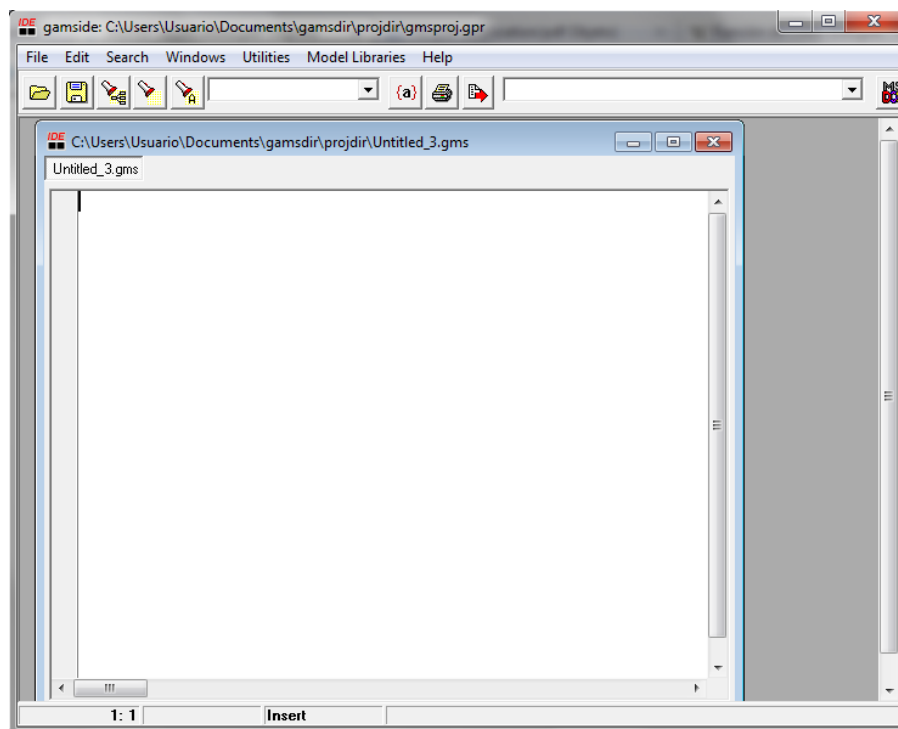
Es necesario resaltar que se pueden trabajar sobre varias ventanas de input y reportes según el número de documentos que se tengan abiertos y el trabajo que se haga con este.



El segundo grupo de ventanas del área son las ventanas relacionadas con los reportes de estado de compilación, estas muestran básicamente el estado que tiene o mantuvo el programa luego de una compilación, el número de errores y una tipificación de ellos, el tipo de licenciamiento del programa y la etapa de simulación con respecto al modelo compilado. Aquí también pueden existir múltiples ventanas.



Para crear un nuevo archivo vamos a File y le damos clic en la opción *New*, de esa forma el programa nos muestra la ventana donde podemos comenzar a editar nuestro modelo:



GAMS permite resolver sistemas lineales, no lineales y enteros, para esto se necesita la creación de un fichero de datos donde el programa recoge todos los datos necesarios para la construcción y resolución del modelo. Para esto es necesario introducir los datos y ecuaciones de una manera ordenada en unos bloques, donde algunos son obligatorios y otros opcionales (dependiendo de la dificultad del modelo).

5. PROGRAMACIÓN

Antes de comenzar a trabajar con cualquier tipo de programación en GAMS es necesario aclarar un conjunto de reglas para la incorporación de valores en GAMS:

1. En GAMS es indiferente el uso de mayúsculas y minúsculas.
2. Todos los comandos deben ser culminados con un punto y coma (;).
3. GAMS reserva un conjunto de palabras para la caracterización de sus valores, en consecuencia los identificadores únicos que el programador designe deben ser diferentes a dichas palabras reservadas.
4. Los identificadores únicos que el programador utiliza para especificar su modelo (nombres de variables, parámetros, modelos...) deben comenzar por una letra y puede contener hasta 10 símbolos alfanuméricos en total, no está permitido el uso de caracteres especiales como la letra ñ, tildes o acentos.

5. Los espacios en blanco consecutivos son comprendidos por GAMS como un único espacio, por lo cual estos pueden ser utilizados a preferencia del usuario para dar que orden que requiera.
6. La única condición de orden para la inserción de bloques y parámetros es la previa declaración de valores y datos.

Adaptado de: Richard E. Rosenthal. “*A GAMS TUTORIAL*”.
Bloque de Ayuda de GAMS.

Todo programa en GAMS esta constituido por un conjunto de bloques que especifican diferentes características del modelo que estemos interesados en simular:

Los bloques obligatorios básicos son:

Bloque	Nombre del bloque
Variables	VARIABLE(S)
Ecuaciones	EQUATION(S)
Modelo	MODEL
Solución	SOLVE

Algunos de los bloques opcionales más utilizados son:

Bloque	Nombre del bloque
Conjuntos	SET(S)
Datos	DATA
Visualización	DISPLAY
Parámetros	PARAMETERS

Adicional a estos bloques, en el cuadro de edición se puede incluir un comentario o una descripción del modelo, esto se logra con una *línea de comentario*.

Las **líneas de comentario** son un espacio donde podemos introducir texto sin que el interfiera con la programación del modelo, muchas veces es importante introducir una línea de comentario para explicar el modelo o el objetivo del modelo. Para introducir un comentario tenemos dos métodos:

- a. Empezar la línea con un asterisco (*), aunque se debe tener en cuenta que no se pueden escribir tildes ni ñ.
- b. Si el usuario necesita escribir mas de una línea de comentarios, se vuelve tedioso escribir varias veces asterisco, en contraste se puede utilizar el comando *\$ontext* para abrir un párrafo de comentarios, y *\$offtext* para cerrarlo, allí podemos incluir la cantidad de líneas que requiramos sin importar su extensión.

Descripción de los bloques básicos:

- **Variables**

En este bloque se definen las variables que se van a usar en el modelo, estas son las variables que va a calcular GAMS con base en los parámetros y la ecuaciones especificadas más adelante. El bloque debe comenzar con la palabra VARIABLE, y se define solo el nombre de las variables, separadas por coma (,) y al final se pone punto y coma (;):

```
variable
t1, t2, q;
```

GAMS acepta una gran variedad de tipos de variables, que pueden ser definidas en distintas formas. Las variables trabajadas permitidas son:

Tipo de variable	Intervalo	Intervalo por defecto
Binary	$\{0, 1\}$	$\{0, 1\}$
Free (default)	$(-\infty, \infty)$	$(-\infty, \infty)$
Integer	$\{0, 1, \dots, n\}$	$\{0, 1, \dots, 100\}$
Negative	$(-\infty, 0)$	$(-\infty, 0)$
Positive	$(0, \infty)$	$(0, \infty)$

Estas se definen con la instrucción *Tipo variables*, por ejemplo:

```
positive variables
X1, X2, P1, P2, V1, V2;
```

Adicionalmente, en cualquier lugar de la programación y según se requiera se pueden imponer condiciones sobre las variables colocando una extensión después del identificados de la variable.

Extensión	Función
.LO (.lo)	Establece un valor mínimo para la variable
.L (.l)	Hace referencia al nivel optimo o real de la variable
.UP (.up)	Establece el nivel máximo que puede tomar la variable
.M (.m)	Hace referencia al valor de cambio marginal de la variable según las condiciones del modelo

Por ejemplo, si se tienen dos variables definidas como Q (cantidades) y P (Precio), de tal forma que se requieran restricciones como la existencia de cantidades y un precio máximo, escribiríamos en GAMS como parámetro complementario:

$$Q.lo=0;$$

$$P.up=15;$$

Asimismo, al momento de realizar consultas, o pedirle a GAMS reportes, podemos especificar el valor de la variable que necesitemos ($display Q.l, p.l$)

o Ecuaciones

En esta parte de la programación se definen las ecuaciones que van a componer el modelo; primero se coloca el nombre de la ecuación seguido de dos puntos (.), luego procedemos a colocar la ecuación, utilizando convenciones propias del lenguaje de programación para hacer las desigualdades o la igualdad, estas son:

- a) =e= para decir que es exactamente igual.
- b) =I= para señalar que es menor o igual que.
- c) =g= para decir que es mayor o igual que.
- d) =It= para expresar que es menor que.

A parte de los signos de relación, en esta parte usamos los operadores básicos como suma (+), resta (-), multiplicación (*), división (/) y exponenciación (**).

Para definir este bloque escribimos la palabra EQUATIONS, escribimos las ecuaciones y finalizamos cada una con un punto y coma (;).

```
equations
q1, q2, q3;
q1.. q =e= hc1 * (t11 - t1);
q2.. q =e= (k/l) * (t1 - t2);
q3.. q =e= hc2 * (t2 - t12);
```

Aunque no solo en este bloque se pueden definir operaciones, existen algunas pautas básicas para introducirlas²:

² Ver <http://www.uv.es/~sala/GAMS/14.PDF> consultada el 12 de octubre de 2010.

Las operaciones simples se pueden introducir directamente con sus símbolos:

Suma (+), resta (-), producto (*), potencia (**) y división (/).

En tanto, algunas operaciones con índices a manera de ejemplo son:

$\sum_{i,j} A_{ij}$ (**SUM (i,j),A(i,j)**)
 $\prod_i a_i^{b_i}$ (**PROD(i,a(i)**b(i))**)
 Max A_i donde A es un conjunto (**SMAX(i,a(i))**)
 Min A_i donde A es un conjunto (**SMIN(i,a(i))**)

Las operaciones con funciones más comunes son:

Valor Absoluto de X (**ABS (X)**)
 Exponente base e, e^x (**EXP(x)**)
 Logaritmo Natural, $\ln x$ (**LOG(X)**)
 Logaritmo base 10 de X, $\log x$ (**LOG10(X)**)
 Máximo de un conjunto no especificado (**MAX(x,y,...,z)**)
 Mínimo de un conjunto no especificado (**MIN(x,y,...,z)**)
 Distribución normal de media X y desviación estándar Y (**NORMAL(x,y)**)
 Potencia de base X y exponente Y, siendo Y entero (**POWER(x,y)**)
 Cuadrado de X, X^2 (**SQR(x)**)
 Raíz Cuadrada de X (**SQRT(x)**)

Dependiendo de los paquetes alternos que sean utilizados y los modos de programación especificados en estos paquetes, las formas de indicar la estructura de las ecuaciones, variables y parámetros que la componen puede variar.

- **Modelo**

En este bloque se definen las ecuaciones que van formar parte del modelo y también se da el nombre del modelo para que GAMS lo reconozca fácilmente. Se empieza escribiendo la palabra MODEL, luego el nombre del modelo, y entre signos de división (/), separados por comas (,), se nombran las ecuaciones, usando el nombre que se les dio en el bloque anterior, si se desean incluir todas las ecuaciones proclamadas es suficiente introducir la opción (/all/). Finalmente como en todos los bloques cerramos con punto y coma (;).

```
model pared /q1, q2, q3/;
```

- **Solución**

En esta parte invocamos el solver mas adecuado para la solución del modelo, existes diversos tipos de solvers incorporados en el lenguaje de GAMS, algunos vienen incorporados con la versión estándar de GAMS y otros son paquetes complementarios que se pueden adquirir con algún costo desde la web de GAMS. En una sección posterior se trataran los más importantes.

Iniciamos este bloque con la palabra SOLVER, luego le indicamos a GAMS el modelo que vamos a resolver, colocando el nombre que le dimos al modelo en el bloque modelo, después le decimos cual solver va a utilizar y con este los parámetros propios del solver, seguido del punto y coma (;) para cerrar.

```
solve pared using NLP minimizing q;
```

A parte de los bloques principales, podemos especificar otros bloques, los cuales, como ya se había mencionado, son opcionales en la línea de comando, estos son:

- *Conjuntos:* Este bloque nos da la opción de crear conjuntos, tales como índices, a los cuales se les asigna un valor. Se empieza con la palabra SET.
- *Datos:* Este no es un solo bloque, más bien es un conjunto de bloques en el cual se definen atributos específicos fijos a los datos, así podemos definir valores base con la función PARAMETERS, tablas con TABLES y escalares con SCALARS. Normalmente se comienza con la palabra DATES.
- *Visualización:* Empezamos poniendo la palabra DISPLAY; en este bloque podemos especificar la forma como queremos visualizar los resultados finales, si no se lo especificamos, quiere decir que vamos a usar la que GAMS trae por defecto.

6. EJEMPLO

Este ejemplo pretende mostrar la forma de inserción de datos con la forma de programación básica de GAMS.

Se trata de la simulación de una economía simple según el modelo Walras-Cassel, las características propias del ejemplo están definidas por:

2 Insumos (Matriz $G_{2 \times 1}$) . Con precios V_j

Condiciones de producción determinadas por la matriz de coeficientes de producción ($B_{2 \times 2}$).

2 Bienes con su precio respectivo P_i

Funciones de Demanda de la Forma: $D_i = 10V_i - 0.5P_n$ $i, n = 1, 2$

```

parameters

B11 cantidad de insumo 1 para producir el bn 1 /1/
B12 cantidad de insumo 1 para producir el bn 2 /2/
B21 cantidad de insumo 2 para producir el bn 1 /3/
B22 cantidad de insumo 2 para producir el bn 2 /2/

G11 Oferta del insumo 1 /15/
G21 Oferta del insumo 2 /20/

variables

X1 cantidades bn 1
X2 cantidades bn 2
P1 Precio bn 1
P2 Precio Bn 2
V1 Precio insumo 1
V2 Precio insumo 2
;

positive variables

X1,X2,P1,P2,V1,V2;

equations

EqF1
EqF2
EqI1
EqI2
Ben1
Ben2
;

EqF1..  x1=E=10*V1-0.5*p1          ;
EqF2..  x2=E=4*v2-0.5*p2          ;
EqI1..  B11*x1+B12*x2=E=G11       ;
EqI2..  B21*x1+B22*x2=E=G21       ;
Ben1..  p1=E=B11*v1+B21*v2        ;
Ben2..  p2=E=B12*v1+B22*v2        ;

model walcas /all/ ;

Solve walcas using cns ;
display x1.l,x2.l,v1.l,v2.l,p1.l,p2.l;

```

Haciendo referencia a la programación, es claro el uso de seis bloques (*parameters*, *variables*, *equations*, *model*, *solve*, *display*) y un criterio complementario para las variables (*positive variables*).

Deben notarse algunas singularidades de programación en este ejemplo con respecto a la descripción previa de los módulos.

En primer lugar hay que notar que el modulo parámetros define un conjunto de objetos que toman los valores especificados entre slash (*/valor/*) y son utilizados luego en el modulo de ecuaciones.

Por otra parte, se define la restricción de rango en las variables (*positive variables*), que aunque se escribe como un bloque adicional, es en realidad una especificación extendida del bloque *variables*.

Finalmente el bloque *model* define el conjunto de variables usando una instrucción generalizada (*/all/*) que toma todas las ecuaciones del la programación para formar el modelo.

7. OUTPUTS Y ERRORES

GAMS tiene dos tipos de outputs, en primer lugar esta la ventana a la que nos referimos anteriormente donde se muestra el estado del procesamiento de GAMS, y en segundo lugar los ficheros .lst que contienen información independiente de cada uno de los programas compilados en GAMS.

La estructura de los ficheros .lst es³:

1. **Eco de impresión (Echo print):** es una copia del archivo de entrada de GAMS, allí se referencian los comandos con un número por línea para poder referenciarlos posteriormente.
2. **Mensajes de Error (Error messages):** en este campo se reportan los errores en la programación, si es que se cometieron, estos son indicados con un signo de dólar (\$) y brindan la ubicación dentro del eco de impresión y una descripción del error.
3. **Mapas de Referencia (Reference maps):** son mapas de referenciación donde se listan las variables parámetro y otros elementos importantes de la programación.
4. **Listado de Ecuaciones (Equation listing):** GAMS muestra las ecuaciones según el las interpreta, este campo tiene por objetivo reportar el problema específico que GAMS afronta luego de despejes, buscando garantizar que se este resolviendo lo que el programador desea.
5. **Estadísticas del Modelo (Model Statistics):** GAMS hace una estadística acerca del las dimensiones del modelo, es la última salida antes de invocar el solver.
6. **Resumen del Solver (Solve summary):** aquí se muestran datos relacionados con el estado funcional del solver y lo que concierne a sus elementos específicos.
7. **Reporte de la Solución (Solution reports):** luego de la ejecución del solver sobre el modelo se brinda una salida con los resultados obtenidos, se busca garantizar los elementos que hace que el solver tome esa solución como el óptimo o los óptimos del modelo.
8. **Resumen de los Reportes (Report summary):** Finalmente GAMS hace una evaluación total del proceso y los reportes anteriores, mostrando el total de errores y los problemas relacionados con el modelo tales como los procesos inviables y los valores indeterminados.

Cada uno de estos elementos es señalado con cuatro asteriscos antes del nombre, por ejemplo:

**** FILE SUMMARY

³Para una más profunda comprensión ver: <http://www.enr.wur.nl/NR/rdonlyres/B3866E0C-C1EF-463E-A420-90BFAB4884BA/83865/GAMSReaderApril20091.pdf>

8. GAMS EN INTERNET

Una importante alternativa que tiene la programación y solución de modelos con GAMS es el soporte en web que brindan webs tales como la de NEOS Solvers, en la cual se puede acceder al servicio de resolución de modelos programados en y con GAMS.

Esta alternativa consiste en enviar la programación de GAMS a dichos servidores para que sean procesados allí, de tal manera que luego se nos envíe un reporte de los resultados, bien sea on-line o por correo.

En ciertos casos, la magnitud de los modelos en cuanto a variables y ecuaciones suele ser descomunal, al punto de llegar a ser limitada por el tipo de licencias con que trabajamos o el tipo de computador en el que las procesamos.

Para acceder a dicho servicio basta con ingresar la página de NEOS y seleccionar el tipo de Solver con el cual se desee procesar el problema (<http://neos.mcs.anl.gov/neos/solvers/index.html>), luego de ello se procede seleccionando el fichero de GAMS con la programación (el archivo de extensión .gms) y enviándolo.

Cabe resaltar que este servicio no cuenta con todos los solvers disponibles en GAMS pero sí con los más importantes.

9. LIBRERIAS

GAMS incorpora en sus archivos un conjunto de librerías que contienen modelos de diversos campos de aplicación programados en GAMS, estos pueden ser considerados como ejemplos o utilizados directamente a partir de la modificación de algunos parámetros que permitan el ajuste a los requerimientos particulares.

Para acceder a este repositorio de modelos tenemos dos alternativas, la primera de ellas es seguir la ruta *File -> Model Library -> Open GAMS Model Library* y proseguir seleccionando el modelo que más se ajuste a lo que necesitemos.

La segunda forma es a través del botón *Model Libraries* de la barra de menús, donde se encuentran varias opciones para acceder, adquirir información y trabajar con estas librerías.

10.DESCRIPCION DE LOS PRINCIPALES SOLVERS Y ALGORITMOS DE RESOLUCIÓN DE GAMS

En este apartado se muestran algunos de los principales solvers de GAMS y sus funciones, se tratan a partir del tipo de problema que puedan solucionar, tal como lo muestran las siguientes tablas:

Opción	Tipo de problema
lp	Programación lineal
nlp	Programación no lineal
dnlp	Programación no lineal con derivadas discontinuas
mip	Programación lineal entera-mixta
rmip	Programación lineal relajada entera-mixta
minlp	Programación no lineal entera-mixta
rminlp	Programación no lineal relajada entera-mixta
mcp	Problemas complementarios mixtos
mpec	Problemas matemáticos con restricciones de equilibrio
cns	Sistemas no lineales acotados

	Solver/Model type availability - 23.3 November 1, 2009											
	LP	MIP	NLP	MCP	MPEC	CNS	DNLP	MINLP	QCP	MIQCP	Stoch.	Global
ALPHAEC								✓		✓		
BARON 9.0		✓	✓				✓	✓	✓	✓		✓
BDMLP	✓	✓										
COIN	✓	✓										
CONOPT 3	✓		✓			✓	✓		✓			
CPLEX 12.1	✓	✓							✓	✓		
DECIS	✓										✓	
DICOPT								✓		✓		
GUROBI 2.0	✓	✓										
KNITRO 6.0	✓		✓				✓	✓	✓	✓		
LINDOGLOBAL 6.0	✓	✓	✓				✓	✓	✓	✓		✓
LGO	✓		✓				✓		✓			✓
MILES				✓								
MINOS	✓		✓				✓		✓			
MOSEK 6	✓	✓	✓				✓		✓	✓		
MPSGE												
MSNLP			✓				✓		✓			✓
NLPEC				✓	✓							
OQNLP			✓				✓	✓	✓	✓		✓
OSL V3	✓	✓										
OSLSE	✓										✓	
PATH				✓		✓						
SBB								✓		✓		
SNOPT	✓		✓				✓		✓			
XA	✓	✓										
XPRESS 20.00	✓	✓							✓			
Contributed Plug&Play solvers												
AMPLwrap	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DEA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Kestrel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tomado de: <http://www.GAMS.com/modtype/index.htm#MATRIX>

LP (Linear Programming)

Este tipo de solver se encarga de resolver problemas de programación lineal, en los cuales se busca maximizar o minimizar una variable desconocida, teniendo en cuenta que el sistema de ecuaciones es un sistema lineal. En la línea de programación colocamos LP en el nombre del solver, y luego ponemos si queremos que minimice o maximice una determinada variable.

GAMS tiene incorporados varios solvers que trabajan de una manera LP, es decir, que resuelven problemas de este tipo con algunas modificaciones. Estos solvers son BDMLP, CPLEX, entre otros.

- **BDMLP:** Es un solver de LP y MIP que viene gratis con cualquier sistema de GAMS. Esta especialmente diseñado para modelos de pequeño y mediano tamaño. A pesar de esto BDMLP esta capacitado para resolver modelos de programación de tamaño razonable, siempre y cuando el modelo no esté correctamente escalado.

BODMLP puede resolver modelos de los siguientes tipos: LP, RMIP y MIP. Por lo tanto se le debe especificar al BDMLP que tipo de modelo va a solucionar, siguiendo el siguiente esquema:

option LP = BDMLP; (ó se le coloca RMIP o MIP)

- **CPLEX:** Este solver de GAMS le permite al usuario combinar la gran capacidad de modelado de GAMS con la efectividad de optimización de Cplex. Cplex esta diseñado para resolver problemas extensos y difíciles, de una manera rápida y con una mínima intervención del usuario, solucionando algoritmos lineales, cuadráticos y problemas de programación lineal entera mixta.

La programación que obedece CPLEX es la siguiente:

option LP = CPLEX; (ó se coloca QCP, MIP, MIQCP, RMIP o RMIQCP)

CPLEX resuelve problemas LP con muchos algoritmos alternativos. La mayoría de los problemas LP se resuelven mejor usando el algoritmo simple dual de Cplex. CPLEX tiene un algoritmo muy eficiente para los modelos de red. Para estos modelos tenemos las siguientes propiedades:

- Cada coeficiente no-nulo es o un -1 o un 1.
- Cada columna que aparece en este modelo tiene exactamente dos entradas distintas de cero, uno con un coeficiente de -1 y el otro con un coeficiente de 1.

Resolver problemas LP requiere un uso de memoria intensivo, por lo que normalmente se

tiene problema con la memoria física al momento de correr LP extensos. Por esa razón, cuando la memoria es limitada, CPLEX automáticamente hace los ajustes, pero estos pueden llegar a influir negativamente en el proceso, por eso es importante estudiar muy bien los procedimientos predeterminados del solver CPLEX.

NLP (No Linear Programing)

Este tipo de Solver se encarga de solucionar sistemas de ecuaciones donde se tenga que maximizar o minimizar una función objetivo, en donde haya varias ecuaciones y algunas restricciones, y una o varias de ellas sean ecuaciones no lineales. Normalmente GAMS nos permite resolver problemas de este tipo con el solver NLP, donde se pone en el bloque SOLVER, el nombre NLP, y luego la función objetivo a maximizar o minimizar.

En GAMS tenemos varios solver especializados que trabajan con este tipo de problemas no lineales; estos son:

- **CONOPT**: Este tipo de solver pertenece a una de las tres familias de los algoritmos que manejan problemas no lineales. CONOPT además cuenta con tres versiones, la vieja CONOPT1 y CONOPT2 y la más nueva y actualizada CONOPT3.

GAMS no tiene automatizado cual es el modelo que aplica perfectamente para problemas no lineales específicos, es por eso que el usuario debe especificarle con cual modelo prefiera trabajar, y esto se hace de la siguiente forma:

option NLP = CONOPT < CONOPT1, CONOPT2 o CONOPT3>

Uno de los métodos de resolución que tiene este modelo es por medio de varias iteraciones, con el fin de encontrar una respuesta lo más cerca posible de la real; por eso es importante saber interpretar el informe de resolución que el solver nos lanza. Normalmente en ese informe de salida, el programa nos muestra qué versión de CONOPT usamos, y seguido a eso nos muestra las iteraciones que hizo y los resultados de cada una de ellas, donde se puede evidenciar que los últimos valores son similares, y eso indica que esa es la respuesta del modelo.

Como este modelo usa iteraciones, es importante tener clara la trascendencia que tiene la especificación de los valores iniciales. El resultado de las iteraciones depende en gran parte de los valores iniciales, ya que estos dan la pauta para la resolución del problema, y como se sabe, los problemas NLP pueden tener muchos valores donde se satisfaga la ecuación, así que es necesario configurar los valores de entrada de la manera más lógica posible.

- **MINOS**: Esta es una adaptación de la versión de un solver que da respuesta a problemas lineales y no lineales en el entorno de GAMS. Está diseñado para encontrar soluciones que son *localmente óptimas*.

MINOS es capaz de resolver problemas de tipo LP, NLP, DNLP y RMINLP. Al igual que otros solvers, es necesario que a MINOS se le especifique el tipo de problema de que se quiere solucionar, esto lo logramos usando en el modelo de GAMS el siguiente comando:

option NLP = MINOS; (ó también se puede colocar lp, dnlp ó rminlp)

También se puede especificar la versión de MINOS con la que desea trabajar poniéndole al lado de `min` el número de la versión, como por ejemplo: `min 5.4`, ya que las versiones pueden tener ciertas diferencias en la resolución de problemas.

- **SNOPT**: (Sparse Nonlinear Optimizer), este solver tiene una modificación importante con respecto a los anteriores vistos, y es que él implementa un proceso que contiene *programación de secuencia cuadrática (SQP)*, donde busca resolver los problemas NLP de optimización usando funciones objetivo y las restricciones de frontera.

Para la programación de este solver en GAMS, ponemos este comando:

option NLP = SNOPT; también puede usarse cambiando nlp por dnlp.

Para este solver toca tener cuidado si el modelo contiene derivadas discontinuas, ya que SNOPT no las reconoce como discontinuas, y ahí se puede generar un error importante.

- **PATHNLP**: Este solver resuelve problemas NLP usando condiciones óptimas de primer orden, donde se asocia con el NLP y resuelve los problemas usando el solver PATH para problemas complejos.

El solver PATH no construye una extensa reducción del hessiano en el espacio, lo que le da ventaja frente a la mayoría de solvers que se encargan de crear códigos de resolución, en realidad, bastante extensos.

Para la introducción de este tipo de solver a GAMS, dado el caso en que no lo tengamos instalado, necesitamos incluir en la programación la siguiente línea de comando:

option NLP = PATHNLP

Si el usuario tiene instalado el PATHNLP, es posible que la mayoría de los problemas NLP se resuelvan por defecto usando este método.

- **LGO**: Este solver trabaja normalmente con modelos no lineales, y cuenta con una suite bastante robusta y eficiente, compuesta varios solvers no lineales. LGO usa como estrategia principal la combinación de búsqueda global y local.

Para lograr que el solver corra en la plataforma de GAMS debemos introducir este texto:

UNIVERSIDAD NACIONAL COLOMBIA
FACULTAD DE CIENCIAS ECONÓMICAS
UNIDAD DE INFORMÁTICA Y COMUNICACIONES

option NLP = LGO; (o LP, RMIP, NLP, o DNLP)

- **MOSEK:** Como tal MONSEK no es exactamente un solver, si no que es un software que contiene la información necesaria para resolver problemas de carácter lineal, no lineal, cuadráticos, entre otros problemas de optimización. Utiliza un algoritmo interior muy eficiente, el cual tiene complejas opciones de solver a partir de las cuales el usuario puede especificarle los parámetros de la optimización teniendo en cuenta el modelo particular que este programando.

Es importante tener en cuenta que para poder utilizar MONSEK se debe tener una licencia de uso, existen tres tipos de licencia de MONSEK:

- ❖ **Solver Link:** Esta diseñado para las personas que prefieran usar MONSEK dentro de GAMS o algunos otros programas.
- ❖ **Base:** Contiene acceso a todos los modelos continuos.
- ❖ **Extendida:** Contiene los mismo modelos que en la licencia base, pero adicionalmente tiene modelos de solución que involucran variables discretas.

DNLP (No Linear Programing with No Continuous Derivate)

Estos problemas son problemas no lineales, pero donde las derivadas no son continuas, es decir, son problemas más complejos que los de tipo NLP, ya que esos problemas se resolvían suponiendo continuidad, y aquí ya no se puede suponer eso. En este tipo de problemas se busca maximizar o minimizar una función objetivo, donde se tenga un sistema que contenga por lo menos una ecuación que no sea lineal con derivadas continuas. Para indicarle a GAMS que use un solver diseñado para este tipo de problemas, simplemente le especificamos en el bloque de SOLVER el nombre del solver, poniendo ahí DNLP, y así el programa ya sabe de que manera procede a resolver los problemas.

Por supuesto para los problemas DNLP, se cuenta con una gran gama de solver especializados dependiendo del modelo, estos son: CONOPT, MINOS, SNOPT, BARON, LGO, OQNLP y MONSEK.

- **BARON:** Es un solver de GAMS que resuelve problemas no lineales de todo tipo. A diferencia de los solvers tradicionales de problemas NLP. El solver BARON implementa un algoritmo de optimización global que actúa bajo parámetros claramente específicos. También implementa algoritmos con una gran cantidad de constantes de propagación y técnicas duales para reducir rango de variabilidad durante el avance del algoritmo.

BARON esta en capacidad de resolver modelos de tipo: LP, MIP, RMIP, NLP, DNLP y

MINLP. Si BARON no esta especificado por defecto, es necesario indicarle a GAMS que queremos resolver el problema utilizando este solver, y lo hacemos de la siguiente manera:

option NLP = BARON;

O en lugar de nlp, podemos colocar el tipo de problema que tengamos.

- **OQNLP:** Es un solver cuyo algoritmo está diseñado para encontrar soluciones globales optimas de sistemas no lineales (NLP), utilizando varios puntos de inicio, y de esta forma consigue ser mucho mas exacto en las respuesta que da; como en la mayoría de los problemas, lo que buscamos con este solver es maximizar o minimizar una función objetivo específica. Esos puntos de inicio son computados por un sistema de implementación llamado OptQuest. Ese sistema tiene la capacidad de manejar variables discretas, por lo tanto hace que se puedan resolver problemas con una o varias variables discretas.

Para utilizar este solver es importante habilitarlo en la ventana de configuración de solvers.

MIP (Mixed Integer Programing)

Los problemas lineales enteros mixtos MIP buscan respuesta a un sistema de ecuaciones donde no solo se manejen enteros, si no que también se manejen números binarios, y al mismo tiempo se maximiza o minimiza una función objetivo. Este tipo de solver es una herramienta importante en modelados que requieran sistemas binarios, ya que los sistemas binarios simplifican la cantidad de variables presentes en un modelo. Con este solver toca tener un poco de cuidado, pues al ser GAMS un modelo profesional, muchas veces no nos da la respuesta exacta pero sí una aproximada, por lo tanto para corregir esto le cambiamos el grado de tolerancia con el siguiente comando:

option OPTCR=0.00001

De esta forma le indicamos que el grado de tolerancia que queremos para el modelo es del 0.00001.

GAMS cuenta además con unos solver que manejan problemas de tipo MIP, estos son: CPLEX, OSL, XA y XPRESS.

- **OSL:** Es un IBM de optimización, el cual tiene incorporado una gran cantidad de solvers dedicados a resolver problemas LP, MIP y QP. Sin embargo GAMS no soporta los métodos de resolución de problemas QP que tiene OSL incorporados, por eso este solver se usa en GAMS únicamente para resolver problemas LP y MIP, ya que para los problemas QP existen otros solvers como CONOPT. OSL permite utilizar diversos

algoritmos de cálculo, y la mayoría de ellos son compatibles con GAMS, por lo que esta es una herramienta comúnmente utilizada en los modelos matemáticos de GAMS.

OSL es capaz de solucionar modelos de tipo LP, RMIP y MIP. Al igual que con otros solvers, para poder trabajar con OSL es necesario indicárselo a GAMS por medio de la siguiente línea de texto:

option LP = OSL; (ó RMIP ó MIP)

Esta línea debe aparecer antes de que se haga el bloque de SOLVER.

- **XA:** Es un solver diseñado para solucionar problemas LP y MIP y cuenta con varios solvers capacitados para resolver problemas del mismo estilo. Este solver usa algoritmos del tipo simplex y barrier, donde con el método simplex consigue buenas respuestas de modelos LP pero la mayoría de veces con procedimientos muy robustos; barrier se estableció para resolver modelos largos. Sin embargo, pese a sus diferencias, ambos algoritmos contribuyen a minimizar el tamaño del modelo, y lograr que las respuestas que da el solver sean bastante buenas.

XA está diseñado para solucionar problemas de tipo LP, RMIP y MIP; sin embargo, es necesario realizar la configuración para que, por defecto, él sea el que soluciona esos modelos. Si no se quiere hacer dicha configuración, se puede introducir en la programación de GAMS la siguiente frase:

option LP = XA; (o MIP o RMIP)

- **XPRESS:** XPRESS es un solver creado para correr únicamente con el sistema de modelación GAMS. Es un sistema versátil de optimización que integra un poderoso solver LP, un módulo MIP, y un módulo barrier para los sistemas LP demasiado largos.

Este solver normalmente se encuentra instalado en GAMS e incluso puede ejecutarse en la versión de demostración (resolviendo sólo modelos pequeños). Si se adquiere una licencia de GAMS y esta incluye XPRESS, este solver no tendrá ninguna limitación de tamaño o de programación.

Este solver está en capacidad para resolver problemas LP, RMIP y MIP, normalmente todos los problemas de estos tipos se resuelven con XPRESS por defecto, sin embargo, si la configuración ha sido cambiada, podemos llamar el solver desde la línea de programación de GAMS, de la siguiente manera:

Option LP = XPRESS; (o MIP o RMIP)

UNIVERSIDAD NACIONAL COLOMBIA
FACULTAD DE CIENCIAS ECONÓMICAS
UNIDAD DE INFORMÁTICA Y COMUNICACIONES

Además de ese comando para llamar al solver, XPRESS cuenta con otras opciones de configuración, que pueden llegar a especificar el modo como él resolverá el modelo de acuerdo con lo que al usuario le convenga.

RMIP (Relaxed Mixed Integer Programming)

Este tipo de problemas es muy similar a los de tipo MIP, pero con algunas diferencias en cuanto al manejo de los enteros, por esta razón, los solvers que se utilizan para estos problemas son los mismos que en MIP.

MINLP (Mixed Integer Nonlinear Programming)

Los modelos MINLP son modelos que combinan aspectos combinatorios con no linealidades. Son mucho más difíciles que los de Programación Lineal Entera Mixta (MIP) y no lineal (PNL) modelos, los solvers que pueden solucionar este tipo de problemas son:

- **AlphaECP** Es un algoritmo basado en el método de técnicas extendidas de corte plano.
- **BARON** Se basa en el algoritmo de ramificación y reducción de N. Sahinidis.
- **DICOPT** El programa se basa en las extensiones de los algoritmos de aproximación externa de la estrategia de la relajación de la igualdad.
- **LOGMIP** LogMIP (acrónimo de lógica de programación entera mixta) es un programa de solución para los programas de disyuntiva o generalizada.
- **LINDOGLOBAL** es un solucionador sustentado en los sistemas de Lindo, Inc. Para usar este solver es necesario contar con una licencia de GAMS/LINDOGLOBAL y GAMS/CONOPT.
- **SBB** algoritmo de rama y acotamiento de Arki.

La forma de usar estos elementos para trabajar con la resolución de problemas de la forma MINLP, al igual que los otros problemas es:

usando la opción de configuración general

Option MINLP = “Nombre Corto del Solver”.

RMINLP (relaxed mixed integer nonlinear programming)

Los problemas no lineales enteros-mixtos relajados son muy similares a los problemas no relajados (MINLP), la única diferencia radica en la restricción de enteros en la solución. En

consecuencia, se pueden usar los mismos solvers que en MINLP simplemente especificando en el modelo las condiciones, por ejemplo:

$$\begin{aligned} & \text{Maximizar o minimizar } f(x) + d(y) \\ & \text{sujeto a } g(x) + H(y) \leq 0 \\ & L < x < U \\ & y > 0 \end{aligned}$$

Usando los elementos de restricción sobre las variables:

Positive variables o *Negative Variables*

x.lo = "valor"; para el valor mínimo de la variable x

x.up = "valor" ; para el valor máximo de la variable x

MCP (Mixed Complementarity Problem)

Los problemas de complementariedad mixta tienen su origen en muchas áreas de aplicación, incluyendo la economía aplicada, la teoría de juegos, ingeniería estructural e ingeniería química.

Constituyen una clase de problemas bastante generales, abarcando los sistemas de ecuaciones no lineales, los problemas no lineales de complementariedad y desigualdades de variación dimensional finita.

- **MILES** es un solucionador de problemas de complementariedad no lineal y sistemas de ecuaciones no lineales que utiliza el método de Newton para la resolución de dichos problemas. Es un solver que viene incorporado en las librerías de GAMS y es recurrentemente utilizado por otros solvers como el MCP.
- **NLPEC** es un solucionador que resuelve problemas del tipo MPEC y MCP mediante transformaciones que los replantean como NLP usando 23 diferentes estrategias re reformulación y recomponiendo los resultados de vuelta a MPEC o MCP.

Existen dos líneas de comandos que permiten establecer la resolución de problemas MCO y MPEC a través de NLPEC, en primer lugar se presenta la formulación de estamento opcional:

option MPEC=NLPEC; (or MCP)

y luego la opción que configura por defecto en GAMS el uso de NLPEC.

GAMS nash MPEC=nlpec MCP=nlpec

- **PATH** es un algoritmo de solución basado en el método de Newton, que incorpora algunas mejoras que permiten generalizar su alcance. El solver PATH esta incluido en las librerías estándar de GAMS y es una alternativa disponible en NEOS (solución web) para la resolución de problemas.

MPEC (Mathematical Programs with Equilibrium Constraints)

Los problemas matemáticos con restricciones de equilibrio son problemas con aplicaciones muy variadas de la forma:

$$\begin{aligned} & \text{Maximize or Minimize } f(x,y) \\ & \text{subject to } g(x,y) < 0 \\ & Lx < x < Ux \\ & F(x,y) \text{ perp-to } Ly < y < Uy \end{aligned}$$

Por defecto el solver que se utiliza en GAMS para la resolución de problemas del tipo MPEC es CONOPT3.

En consecuencia, los problemas del tipo NLP y MCP están incorporados como casos particulares de MPEC.

Los solver general para este tipo de problemas es NLPEC descrito en el modulo anterior, sin embargo hay una alternativa importante en el campo de los desarrollos económicos, siendo MSPGE una importante alternativa⁴.

CNS (Constrained Nonlinear System)

GAMS introduce un nuevo tipo de problema denominado CNS, los Sistemas no lineales limitados dos sistemas de ecuaciones no lineales cuadrados, es decir, con igual número de ecuaciones al de variables.

El uso de restricciones con desigualdades no implica un incumplimiento de las normas establecidas, sin embargo, hay que tener en cuenta que la naturaleza del problema genera resultados únicos para las variables, valores de nivel constantes, por lo que la columna de variación marginal en los reportes de GAMS no incluirán este valor, asimismo la petición de estos valores en la programación de GAMS generaran informes vacios.

Son dos los solvers usados para la resolución de problemas del tipo CNS en GAMS: **PATH** y **CONOPT** en cualquiera de sus tres versiones.

⁴ En este manual se dedica un modulo completo a al a descripción de estas librerías.

Cabe señalar que cada uno de los solvers aquí descritos están limitados por la disponibilidad de las distribuciones para diferentes plataformas. La siguiente tabla resume la disponibilidad de los solvers por distribución de GAMS.

Supported Platforms

	Solver/Platform availability - 23.5 July 4, 2010													
	x86 MS Windows	x86_64 MS Windows	x86 Linux	x86_64 Linux	Sun Sparc SOLARIS	Sun Sparc64 SOLARIS	Sun Intel SOLARIS	IBM RS 6000 AIX 5.3	Mac Intel32 Darwin	Mac x86_64 Darwin	HP 9000 HP-UX 11 ¹	SGI IRIX ²	DEC Alpha Digital Unix 4.0 ³	Mac PowerPC Darwin ⁴
ALPHAECF	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
BARON 9.0	✓	✓	✓	✓										
BDMLP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
COIN-OR	✓	✓	✓	✓										
CONOPT 3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CPLEX 12.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	10.0	9.1	8.1	
DECIS	✓	✓	✓	✓	✓	32bit					✓	✓	✓	
DICOPT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
GUROBI 3.0	✓	✓	✓	✓						✓				
KNITRO 6.0	✓	✓	✓	✓	5.2	32bit			✓					
LINDOGLOBAL 6.1	✓	✓	✓	✓	6.0	6.0	✓		✓					✓
LGO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MILES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MINOS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MOSEK 6	✓	✓	✓	✓	5.0	5.0	✓	✓	✓	✓	3.2	✓	✓	✓
MPSGE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSNLP	✓	✓	✓	✓	✓	32bit			✓	✓	✓	✓	✓	✓
NLPEC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
OQNLP	✓	32bit	✓	32bit										
PATH	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SBB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCIP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SNOPT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
XA	✓	32bit	✓	✓	✓	32bit					✓	✓	✓	✓
XPRESS 20.00	✓	✓	✓	✓	✓	✓	✓				16.10			

¹GAMS distribution for HP 9000/HP-UX is 22.1.
²GAMS distribution for SGI IRIX is 22.3.
³GAMS distribution for DEC Alpha is 22.7.
⁴GAMS distribution for Mac PowerPC is 23.3.

Contributed Plug&Play solvers														
AMPLwsp			✓		✓		✓		✓		✓		✓	✓
DEA			✓		✓		✓		✓		✓		✓	✓
Kestrel			✓		32bit		✓		32bit		✓		✓	✓

For backward compatibility we maintain older versions of operating systems and solvers. Please call.

Tomado de: <http://www.GAMS.com/solvers/index.htm#platforms>

11. PAQUETES COMPLEMENTARIOS PARA CIENCIAS ECONÓMICAS (MPSGE/GAMS)⁵

MPSGE (*mathematical programming system for general equilibrium analysis*) es un sistema de programación matemática enfocado a la especificación de modelos de equilibrio general computable, opera como un subsistema de GAMS y esta habilitado en todas sus distribuciones aunque tiene una estructura de planteamiento y sintaxis de programación diferente que se aproxima más a los requerimientos del tipo de problemas que plantea y a la convenciones trabajadas por los economistas en simulaciones recurrentes en el tiempo y de optimización en

⁵ La descripción y los procedimientos de MPSGE/GAMS señalados en este apartado son basados en la guía de usuario del paquete "MPSGE: A User's Guide" de James Markusen y Thomas Rutherford de 2004.

escala de modelos amplios en magnitud.

El usar una estructura diferente a la acostumbrada algebra de conjunto de ecuaciones permite por una parte centrar la atención en los resultados de la simulación (los valores y dinámicas de las variables) además de reducir el tiempo de programación al mínimo, sin embargo impide la visualización transparente de la estructura del modelo (la forma misma de las ecuaciones).

Las librerías de MPSGE son fuertes en el manejo de funciones no lineales del tipo CES contribuyendo a la comprensión de los usuarios con poca experiencia en programación extensiva y acostumbrada a las convenciones de la teoría económica.

Elementos de un modelos MPSGE

Los modelos de equilibrio general computado pueden reducirse a tres conjuntos básicos:

p (precios) no negativos, con una matriz de los n productos (bienes finales, intermedios y factores primarios de producción).

y (producción) no negativa en cada uno de los m sectores de la economía.

m (ingresos), una matriz de pagos a cada uno de los agentes de la economía simulada.

A su vez, las condiciones de equilibrio se especifican sobre estos tres elementos y la correspondencia entre las variables.

Las Características de un Modelo Estándar de Equilibrio General son:

1. Interacción de múltiples agents.
2. Decisiones individuales basadas en la optimización
3. La interacción de los agentes es mediada por las dinámicas del Mercado y los precios
4. El equilibrio se da cuando las variables endógenas (Precios), se ajustan tal que
 - Los agentes sujetos a sus restricciones no pueden mejorar sus decisiones (los productores maximizan el beneficio y los consumidores maximizan su utilidad).
 - Los mercados generalmente se limpian. Las cantidades ofrecidas igualan a las cantidades demandadas en cada Mercado.

Condiciones de Equilibrio:

Precio de las Mercancías y Vaciamiento del Mercado: se especifican las condiciones de tal manera que el precio de las mercancías garanticen la compatibilidad entre oferta y demanda obteniendo como resultado el vaciamiento del mercado.

Nivel de Actividad Productiva y **Beneficio Nulo**: las tecnologías de producción con que cuenta la oferta deben ser consistentes con el supuesto de beneficio cero en la estructura de competencia perfecta.

Nivel de Ingresos de los Consumidores y **La Definición de Esos Ingresos**: se establece la reciprocidad y equivalencia entre el consumo realizado por los agentes de la economía y el origen de esos ingresos.

Cada una de estas condiciones debe ser atribuida en base al principio de complementariedad económica.

A partir de estas condiciones se pueden establecer un conjunto de instrucciones para la construcción de los modelos MPSGE:

1. Recolección de datos de origen, normalmente una tabla de I-O, la encuesta de hogares y un conjunto de estadísticas de comercio. Estos datos se obtienen a menudo en formato XLS.
2. Interpretación, la reconciliación y el equilibrio de los datos de origen.
3. Especificación de las variables del modelo y las ecuaciones.
4. Calibración de los coeficientes de las preferencias y la tecnología.
5. La replicación del equilibrio de referencia para verificar que las ecuaciones del modelo y la derivada de los coeficientes son internamente consistentes.
6. Definición de escenarios y la solución de los escenarios.
7. La generación de informes (tablas y figuras)
8. La interpretación y la escritura.

11.1. *Sintaxis con GAMS/MPSGE*

La estructura típica de un programa en GAMS/MPSGE es:

1. Declaración de conjuntos
2. Declaración de parámetros y definiciones
3. Definición del Modelo
4. Invocación del modelo
5. Definición de los escenarios de solución

Donde los elementos 1 y 2 hacen referencia a al establecimiento de variables, parámetros y elementos que hacen parte del modelo y problema a trabajar.

La definición del modelo se realiza en un bloque delimitado por **\$ontext** y **\$offtext** sin que este sea un conjunto de líneas de comentario, en realidad los comentarios en GAMS/MPSGE se notan a partir del carácter "!", en este bloque se definen las características de los agentes y las interacciones existentes en el mercado.

La invocación del modelo se hace inmediatamente terminado el bloque de definición del modelo con una línea como esta: **\$sysinclude mpsgeset modelID**, donde **modelID** hace referencia a un id del tipo de modelo que se esta trabajando, es el nombre que se le asigna al modelo.

La definición de los escenarios de solución y solución se plantea en dos etapas, en primer lugar la reproducción del problema o reproducción de la situación inicial que se realiza con el comando: **\$include modelID.gen**. Y en segundo lugar se define el modelo a solucionar y el solver, sugiriendo a demás la modificación en niveles propios de las simulaciones, se realiza de la misma manera que en el lenguaje generalizado de GAMS/MPSGE, así: **solve modelID using mcp**.

PALABRAS CLAVE DE LA PROGRAMACIÓN EN GAMS/MPSGE

Hay nueve palabras clave MPSGE que aparecen en un modelo de declaración MPSGE. Estas son:

\$MODEL: modelID Esta sentencia asigna un identificador al modelo actual. Esta debe ser la primera instrucción dentro del bloque de **\$ONTEXT- \$OFFTEXT**. MPSGE trata a todos los registros que aparecen antes de la palabra clave **\$MODELO** dentro del bloque de **\$ONTEXT- \$OFFTEXT** como comentarios, **modelID** debe ser a la vez un modelo legítimo GAMS identificador y un nombre de archivo válido. Este nombre se utiliza para formar **modelID.GEN**. (Este nombre de archivo debe estar en mayúsculas cuando se ejecuta en Linux o UNIX).

\$SECTORS:, \$COMMODITIES:, \$AUXILIARY:, \$CONSUMERS: Estas cuatro palabras definen las variables que se utilizan para definir el modelo MPSGE. Las entradas de estos bloques comparten la misma sintaxis.

La palabra clave **\$SECTORS** no aparece en los modelos sin producción (economías de intercambio puro).

La palabra clave **\$AUXILIARY** sólo se utiliza en los modelos con las limitaciones de los impuestos y endógenos o las restricciones.

\$PROD:sector Esta declaración indica el inicio de un conjunto de registros que definen la tecnología y los impuestos para una actividad en particular.

\$DEMAND:consumer Esta declaración indica el inicio de un conjunto de registros que

definen dotaciones iniciales y las preferencias de los consumidores en el modelo. La representación de las preferencias y tecnología de seguir una sintaxis similar, basado en la forma de compartir calibrado funciones CES encadenadas.

\$REPORT: Esta declaración indica el inicio de un conjunto de registros que definen las variables a través del cual GAMS MPSGE devuelve insumos de producción, los productos de productores y / o demanda de los consumidores. La variables definidas en bloques de \$ informe no puede ser utilizado en las ecuaciones del modelo.

\$CONSTRAINT:auxiliary Esta declaración específica la condición de equilibrio que ha de estar asociado con una variable auxiliar en particular. Condiciones de equilibrio se escriben después de las reglas convencionales para la expresión de las ecuaciones en GAMS. Unas restricciones auxiliares pueden incluir variables que corresponden a los precios de los productos básicos, los niveles de producción de la actividad y los ingresos de los consumidores.

FUNCIONES

La forma de las funciones debe ser declarada sobre los agentes que interactúan en la economía, la oferta y la demanda (productores, consumidores y gobierno). Por defecto todas las funciones se trabajan a partir de una función CES, donde los parámetros se definen inmediatamente después de declarar el agente, y la incorporación y características de las variables en las líneas siguientes con el formato mostrado a continuación:

I: Inputs

O: Outputs

Q: Cantidad de referencia. El valor por defecto es 1.

P: Precio de referencia. El valor por defecto del precio de referencia es de 1.

R: Impuesto de renta. Este campo debe ser seguido de un nombre de los consumidores.

T: Tasa de impuesto campo identificador. (Más de un impuesto se aplique a una sola entrada.)

N: El impuesto sobre endógeno. Esta etiqueta debe ser seguida por el nombre de una variable auxiliar.

M: El impuesto sobre endógeno multiplicador.

Ejemplo:

Bloque de producción definido por una función Cobb-Douglas (CES de semi-elasticidad unitaria $s=1$).

```

$PROD:Y(r) s:1
O:P(r)      Q:y0(r)
I:W(f,r)    Q:x0(f,r) P:w0(f,r)
    
```

EJEMPLO DE PROGRAMACIÓN EN GAMS/MPSGE

Tomado del Curso de Equilibrio General Computable desarrollado en la FCE.

```

$ontext
Let us introduce the govt. It only collects an income tax and the tax revenue
is fully spent to buy goods and services (public expenditures). Government budge
is to be balanced (a conservative rule....)

          ! Production Sectors ! Agents
          ! Cost              ! CONS  INV  GOVT  !
-----
Sales      ! 100              -75  -15  -10
SAV/INV    !                  ! -15  +15
Wage incomes ! -40              ! 40
Capital (markup) incomes ! -60              ! 60
Income tax  !                  -10              10

$offtext

parameter
|VZ      Real GDP in the benchmark
|KZ      Capital demand in the benchmark
|LZ      Labour demand in the benchmark
|RCONSZ  Real consumption in the benchmark
|RGEXPZ  Real govt expenditures in the benchmark
|t       Average income tax rate
|RGSAVZ  Real govt savings in the benchmark
|INV     Real investment (savings) in the benchmark
|LSZ     Labour supply in the benchmark
|KSZ     Capital supply in the benchmark
|beta    Minimum wage parameter
|tt      TASA 2
;
VZ = 100 ;
KZ = 60 ;
LZ = 40 ;
RCONSZ = 75 ;
INV = 15 ;
LSZ = 40 ;
KSZ = 60 ;
beta = 1 ;
RGEXPZ = 10 ;
RGSAVZ = 0 ;
t = 10/100 ;

$ONTEXT
$MODEL:neoc
$COMMODITIES:
      QV      ! Price index for commodity V (value added)
      r       ! User cost of capital
      w       ! Wage rate
$SECTORS:
      V       ! Activity level for sector V
$CONSUMERS:
      CONS    ! Consumer Expenditures
      GOVT    ! Govt Expenditures
$AUXILIARY:
      ur      ! Unemployment rate
    
```

```

$PROD:V s:1

      O:QV      Q:VZ
      I:r       Q:KZ
      I:w       Q:LZ

$DEMAND:CONS

      D:QV      Q:RCONS
      E:QV      Q:(-inv)
      E:w       Q:(LSZ*(1-t))
      E:w       Q:(-LSZ*(1-t))      R:ur
      E:r       Q:(KSZ*(1-t))

$DEMAND:GOVT

      D:QV      Q:RGEXPZ
      E:w       Q:(t*LSZ)
      E:w       Q:(-t*LSZ)          R:ur
      E:r       Q:(t*KSZ)
      E:QV      Q:(-RGSAPV)

$CONSTRAINT:ur

      w =G= beta*QV ;

$REPORT:

      V:RCON      D:QV      DEMAND:CONS
      V:RGDP      O:QV      PROD:V
      V:LD        I:w       PROD:V
      V:KD        I:r       PROD:V
      V:RGEXP     D:QV      DEMAND:GOVT

$offtext
$sysinclude mpsgeset neoc

* It is possible to fix the numeraire

QV.FX = 1 ;

$INCLUDE neoc.GEN
SOLVE neoc USING MCP;

*Beta = 1.1 ;
inv = 1.15*INV ;
*BETA = 1.1 ;
*T = 1.2*T ;

$INCLUDE neoc.GEN
SOLVE neoc USING MCP;

```

El bloque demarcado por la línea punteada de color negro es la definición del problema, es un párrafo de comentarios que tiene por objetivo resumir el modelo en el estado inicial (benchmark).

El bloque demarcado con la línea de color azul claro hace referencia a la declaración de parámetros y definiciones, la declaración de conjuntos no es necesaria en el planteamiento de este problema.

El bloque demarcado con la línea roja punteada corresponde a la definición del modelo, y el de color verde incorpora las líneas que invocan el modelo y definen los escenarios de solución, para el ejemplo se definen varias simulaciones sobre los parámetros de las funciones y la variable inv.

12. CONCLUSIONES

- ▶ Las características de GAMS permiten al usuario un manejo global de los problemas de optimización otorgándole la posibilidad de centrar su análisis en los resultados del proceso más que en este mismo.
- ▶ La flexibilidad y capacidad de ajuste a las necesidades particulares de cada usuario que ofrece GAMS le perfilan como un software muy robusto en comparación con otros que cumplen funciones similares, facilitando su acogida entre los profesionales y estudiantes de las ciencias económicas.
- ▶ El programa GAMS es una potente herramienta de simulación y optimización altamente utilizada en el mercado por lo que se justifica la oferta de un curso en el manejo de GAMS dirigido a los estudiantes de la FCE.

13. BIBLIOGRAFÍA

Dellink, Rob (2009). *GAMS for environmental-economic modelling*. Extraído el 29 de octubre de 2010, de <http://www.enr.wur.nl/NR/rdonlyres/B3866E0C-C1EF-463E-A420-90BFAB4884BA/83865/GAMSReaderApril20091.pdf>.

GAMS Development Corporation (Segundo Semestre del 2010). *GAMS Home Page*. <http://www.gams.com>.

NEOS. *NEOS Solvers*. <http://neos.mcs.anl.gov/neos/solvers/index.html> consultada el segundo semestre del año 2010.

Markusen, James and Rutherford, Thomas (2004). *MPSGE: A User's Guide*. University of Colorado, Department of Economics. UNSW Workshop.

Richard E. Rosenthal. *A GAMS TUTORIAL*. Extraído el 29 de octubre de 2010, de <http://www.gams.com/dd/docs/gams/Tutorial.pdf>.

Sala Garrido, Ramón. *Spanish Reference Manual*. Extraído el 12 de octubre del 2010, de <http://www.uv.es/~sala/gams/14.PDF>.