

881201
7
29.

UNIVERSIDAD ANAHUAC

ESCUELA DE ACTUARIA

CON ESTUDIOS INCORPORADOS A LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO



MODULO DE SIMULACION MONTE CARLO PARA EL SISTEMA DE ANALISIS INTERACTIVO

(S. A. I.)

TESIS CON
FALLA DE ORIGEN

T E S I S

QUE PARA OPTAR POR EL TITULO DE:

A C T U A R I O

P R E S E N T A

EDUARDO ANTONIO HERNANDEZ ALBIN

MEXICO, D. F.

1987



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

CAPITULO I Sistema de Analisis interactivo.	3
1.1 Sintaxis General del Sistema.	3
1.2 Archivos de Comandos.	5
1.3 Variables.	5
1.4 Bancos de Información.	6
1.4.1 Bancos de Datos.	6
1.4.2 Bancos de documentación.	6
1.5 Estructura del Sistema.	7
1.5.1 Módulos de SAI.	8
1.6 Expresiones Aritméticas.	10
1.6.1 Operadores.	10
1.6.2 Funciones.	11
1.7 Auxilio del Sistema al Usuario.	13
1.7.1 Ayudas.	13
1.7.2 Forma conversacional.	14
1.7.3 Control de Errores.	14
1.8 Referencias.	15
CAPITULO II El Método de Simulación Monte Carlo	16
2.1 Modelos.	16
2.2 Simulación.	17
2.3 Método Monte Carlo.	19
2.3.1 El método Monte Carlo para problemas estocásticos.	21
2.3.2 El método Monte Carlo para resolver problemas determinísticos.	23
CAPITULO III Módulo de Simulación Monte Carlo.	24
3.1 El Módulo de Monte Carlo como parte del Sistema de Analisis interactivo (SAI).	24
3.2 Descripción general de los modelos en el módulo de Monte Carlo.	27
3.3 Fases del sistema.	27
3.4 Creación de un Modelo en Monte Carlo.	28
3.4.1 Escenario de un Modelo.	28
3.4.2 Nombre de un Modelo.	28
3.4.3 Tiempo de un Modelo.	29
3.4.4 Número de Corridas de un modelo.	29
3.4.5 Estadísticas de las variables endógenas de un modelo.	30
3.4.6 Ecuaciones del modelo.	30
3.4.6.1 Operadores.	31
3.4.6.2 Funciones matemáticas.	31
3.4.6.3 Funciones lógicas.	31
3.4.6.4 Funciones estocásticas.	32
3.4.6.5 Otras funciones.	32
3.4.6.6 Ejemplos de ecuaciones escritas para Monte Carlo.	34
3.4.7 Variables atrasadas.	34

3.4.8	Tiempo de Validez de una Ecuación.	35
3.4.9	Funciones definidas por el usuario.	37
3.4.10	Compilación de un modelo en Monte Carlo.	39
3.5	Recuperación de los resultados.	41
3.6	Fase datos (análisis de sensibilidad).	42
3.7	Administración de Ecuaciones y Funciones.	43
3.8	Diagnóstico.	44
3.9	Límites del sistema.	45
CAPITULO IV Generadores de Números Aleatorios.		47
4.1	Generadores de números Aleatorios con distribución uniforme en [0,1].	47
4.2	Generación de números aleatorios con cierta distribución de probabilidad a partir de una sucesión distribuida uniforme en el intervalo [0,1].	49
4.2.1	El método de la Transformada Inversa.	50
4.3	Métodos Alternativos para generación de números aleatorios a partir de sucesiones de números aleatorios con distinta distribución.	51
4.3.1	Método de Composición.	51
4.3.2	Método de Aceptación y Rechazo.	53
4.3.3	Métodos Particulares.	55
CAPITULO V Pruebas de Bondad de Ajuste y Aleatoriedad. ...		57
5.1	Función de Distribución Empírica.	58
5.2	Prueba de Bondad de Ajuste Kolmogorov-Smirnov.	60
5.3	Resultados de las pruebas en los generadores del módulo de Monte Carlo.	62
5.3.1	Rutina para prueba de bondad de ajuste Kolmogorov-Smirnov de IMSL.	62
5.3.2	Tablas de resultados.	63
5.3.3	Conclusión de las pruebas.	66
5.4	Aleatoriedad.	66
CAPITULO VI Ejemplo Práctico.		67
6.1	Introducción.	67
6.2	Modelo de Inventario.	67
6.3	Definición del Modelo en SAI.	69
6.3.1	Problemas de implementación del modelo.	69
6.3.2	Variables del modelo.	70
6.3.2.1	Variables endógenas.	70
6.3.2.2	Variables exógenas.	71
6.3.2.3	Parámetros.	71
6.3.2.4	Ecuaciones en SAI para el modelo.	71
6.3.3	Ejemplo de una corrida de simulación.	73
6.3.4	Análisis de sensibilidad.	76
6.3.5	Conclusiones.	77
CAPITULO VII Conclusiones.		78
APENDICES.		80

8.1 Apéndice A	80
8.1.1 Comandos y atributos generales del sistema.	80
8.1.1.1 Comando NOMBRE.-.....	80
8.1.1.2 Atributo FASE.-.....	81
8.1.1.3 Atributo TIEMPO.-.....	81
8.1.1.4 Atributo CORRIDAS.-.....	82
8.1.1.5 Comando DESPLIEGA.-.....	82
8.1.1.6 Comando BORRA.-.....	83
8.1.2 Comandos y atributos específicos de la Fase	
Completa.	83
8.1.2.1 Atributo ECUACION.-.....	83
8.1.2.2 Atributo FUNCION.-.....	84
8.1.2.3 Atributo ESTADISTICAS.-.....	84
8.1.3 Comandos y atributos específicos de la Fase	
Datos.	85
8.1.3.1 Atributo MODIFICA.-.....	85
8.1.4 Comandos y atributos específicos de la Fase	
Resultados	86
8.1.4.1 Atributo MANTEN.-.....	86
8.2 Apéndice B.- Funciones generadoras de números	
aleatorios.	86
8.2.1 Distribución Gama.	86
8.2.2 Distribución Beta	87
8.2.3 Distribución Binomial.	87
8.2.4 Distribución Cauchy.	87
8.2.5 Distribución Ji-Cuadrada.	88
8.2.6 Distribución Geométrica.	88
8.2.7 Distribución Exponencial.	88
8.2.8 Distribución Poisson.	89
8.2.9 Distribución Normal	89
8.2.10 Distribución Triangular.	90
8.2.11 Distribución Uniforme.	90
8.3 Apéndice C.- Ejemplo de una sesión de un usuario en	
el módulo de Monte Carlo.	91
8.3.1 Sesión de la definición del modelo de	
inventarios.	91

B I B L I O G R A F I A 94

INTRODUCCION

Con el advenimiento de las computadoras de alta velocidad de cálculo y bajo costo de operación, los tomadores de decisiones tienen a su alcance, cada vez más herramientas de análisis que pueden apoyar eficazmente el proceso de planeación. Con éstas, pueden utilizar y evaluar rápidamente la información con la que cuentan, desde los diferentes puntos de vista que ofrecen la Estadística, la Investigación de Operaciones y la teoría de Control, entre otras, reuniendo así, más elementos que ayuden a tomar mejores decisiones.

Hoy día, para que un país o empresa pueda desarrollarse y crecer, es importante que tenga una buena planeación, la cual de llevarse a cabo, le permitirá utilizar los recursos con los que cuenta de manera óptima y racional.

Siguiendo esta estrategia, la Oficina de Asesores del C. Presidente de la República decidió desarrollar un sistema que apoyara a sus economistas en la tarea de analizar la economía de México. El resultado fué el Sistema de Análisis Interactivo (SAI).

El sistema debía ser muy sencillo de utilizar para que fueran los propios economistas de la Presidencia quienes lo usaran. Esto con la finalidad de responsabilizarlos en la realización de sus estudios, limitando la posibilidad de error por malos entendidos con algún personal de sistemas. Para que el sistema fuera realmente útil, debería ser capaz de realizar una gran cantidad de los procedimientos que el analista de procesos económicos lleva a cabo.

SAI resultó ser un sistema muy amigable para el usuario nuevo, pero versátil y poderoso para el más diestro en su manejo. El sistema está diseñado para el manejo masivo de información económica y sus capacidades cubren las diferentes etapas del proceso del análisis de la información: Administración, mantenimiento y consulta, análisis por medio de modelos y filtros estadísticos, y por último, presentación y publicación de los resultados.

Con la finalidad de complementar las herramientas de análisis existentes en SAI, la Presidencia de la República decidió en 1983 incorporar al sistema la técnica de simulación conocida como el Método de Monte Carlo.

Monte Carlo realiza la simulación de modelos matemáticos en los cuales algunas de sus variables son aleatorias. Estas deben pertenecer a una población conocida. Los valores de estas variables son escogidos aleatoriamente por medio de generadores de números aleatorios.

La técnica Monte Carlo es cada vez más recurrida por los tomadores de decisiones de diversos campos. Es muy utilizada para la evaluación de proyectos de inversión, modelos de Investigación de Operaciones, Economía y Finanzas, y para resolver algunos problemas matemáticos, como el cálculo de integrales complicadas.

Es así como la Oficina de Asesores del C. Presidente de la República creó el Módulo de Simulación Monte Carlo para el Sistema de Análisis Interactivo (SAI).

Monte Carlo está construido bajo la misma filosofía de operación y programación que en el resto del sistema. El módulo respeta el mismo estilo de instrucciones que ya existía en SAI. Además, comparte los mismos bancos de datos, lo que permite al analista que desea trabajar con el método de Monte Carlo, utilizar la infraestructura de todo el sistema.

El módulo de Monte Carlo trata de ser lo más general posible para que el usuario logre simular modelos con características muy distintas.

Actualmente, este módulo, junto con SAI, se encuentran operando en el Secretariado Técnico de Gabinetes de la Presidencia de la República, en el Centro de Estudios Económicos de Banamex y en el Instituto Tecnológico Autónomo de México (I.T.A.M.).

La finalidad de este trabajo es dar a conocer al lector la filosofía general de operación del Módulo de Simulación Monte Carlo, así como la del resto del Sistema de Análisis Interactivo. También es describir, las técnicas utilizadas para la generación de las series de números aleatorios, y las pruebas estadísticas realizadas para comprobar su bondad de ajuste y por último, presentar un ejemplo de un modelo ilustrativo de las capacidades que tiene el sistema en la simulación.

CAPITULO I Sistema de Analisis interactivo.

El sistema de analisis interactivo (SAI) es un sistema desarrollado inicialmente en la Oficina de Asesores del C. Presidente de la República en el año de 1979. Actualmente se opera y se sigue su desarrollo en la Dirección General del Secretariado Técnico de Gabinetes de la Presidencia de la República. El sistema reside actualmente en una máquina PDP 11/34 de Digital y está formado de un conjunto de programas que se comunican e interactúan. Su propósito es el de dar apoyo computacional a los analistas e investigadores de la Dirección General y del Comité de Asesores Económicos de la Presidencia.

Actualmente SAI sirve de apoyo a una serie de instituciones ajenas a la Presidencia, como PEMEX y la Secretaría de Hacienda y Crédito Público. Además SAI se encuentra instalado en la computadora VAX 780 del Departamento de Estudios Económicos de BANAMEX, donde presta una gran ayuda a sus investigadores. Asimismo, se encuentra a disposición del Instituto Tecnológico Autónomo de México (ITAM) una versión de SAI que corre en su computador PDP 11/40.

SAI opera interactivamente mediante el empleo de un lenguaje conversacional basado en el idioma español. En cualquier nivel de operación el usuario puede pedir ayuda al sistema sobre las instrucciones y funciones que existen en él. Todo esto hace al sistema tan fácil de usar que inclusive personas que no tienen ninguna formación computacional lo usan sin mayor esfuerzo.

El sistema proporciona al analista principalmente la posibilidad de almacenar información numérica, procesarla y analizarla. Los resultados pueden presentarse inmediatamente mediante reportes gráficos y cuadros numéricos en forma rápida y sencilla. En el diseño de SAI se enfatizó primordialmente su modularidad y su capacidad de crecimiento sin perder el concepto de un solo sistema integrado, totalmente intercomunicado y basado en un mismo estilo de instrucciones, convenciones y secuencia de fases de los distintos procesos. SAI tiene definidas un grupo de instrucciones básicas que permiten resolver una gran cantidad de problemas de análisis, siempre y cuando el usuario especifique explícitamente la secuencia de operaciones que se debe llevar a cabo sobre la información. Sin embargo su desarrollo está orientado principalmente al análisis económico por lo que la información de este tipo se maneja con mayor naturalidad.

1.1 Sintaxis General del Sistema.

El lenguaje de SAI es muy sencillo y por medio de él, el usuario especifica las instrucciones necesarias para que el sistema haga lo que él desea llevar a cabo. Estas instrucciones están formadas de comandos, atributos y argumentos.

Una instrucción comienza por un comando. Este define el tipo de acción que se ejecuta con una instrucción. Por ejemplo el comando "BANCO" especifica que la acción esta relacionada con el manejo de un banco.

Los atributos dan información específica sobre la ejecución de una instrucción, o en ciertos casos designan parámetros cuyos valores están dados por los argumentos. Los atributos pueden ser varios, se separan entre ellos por el signo de ";" y se dan en cualquier orden.

Los argumentos son valores de parámetros designados por un comando o atributo. Estos pueden ser números, expresiones aritméticas, variables o nombres de archivos, dependiendo del comando o atributo al que se asocian.

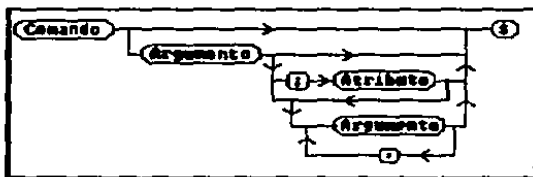
El sistema permite a sus usuarios dar una instrucción en más de un renglón y así como también, dar más de una instrucción en un mismo renglón. Es por esto que es necesario decirle al sistema en donde acaba cada una de las instrucciones. La forma de indicar esto es por medio del signo de pesos "\$". Así, una instrucción en SAI casi siempre termina con este signo. Existen excepciones a esta regla que veremos más adelante.

Ejemplo de una instrucción en SAI es:

BANCO A1;CREA\$

donde "BANCO" es el comando que define que la acción está relacionada con el manejo de un banco de datos, "A1" es un argumento que constituye un nombre de archivo y "CREA" es el atributo que especifica que se trata de la creación de un banco. Por lo que al ejecutar esta instrucción en SAI se crea un banco de datos cuyo nombre es A1.

La sintaxis general de una instrucción en SAI se puede representar por el siguiente diagrama:



Todos los comandos y atributos de SAI tienen que ser especificados cuando menos por sus tres primeras letras. Así por ejemplo, las siguientes tres formas de dar el comando BANCO son equivalentes:

BAN
BANC
BANCO

1.2 Archivos de Comandos.

El analista al utilizar SAI puede dar las instrucciones una por una a través de su sesión de trabajo y el sistema las ejecuta inmediatamente. Sin embargo, existe la posibilidad que utilice archivos creados fuera de SAI que contengan instrucciones para que se ejecuten en el sistema. Estos archivos que llamamos "Archivos de Comandos", pueden ser utilizados por un usuario las veces que quiera e incluso puede crear uno de tal forma que al ejecutarlo pregunte por algún atributo o valor específico de un parámetro.

El comando para mandar ejecutar un archivo de comandos es " @ " y tiene como argumento el nombre con el que se creó el archivo. Por ejemplo, si un usuario tiene un archivo de comandos llamado GRAF1 la instrucción que debe proporcionar es: @ GRAF1

(Nota: Este es una de las pocas instrucciones que no terminan con el signo de "\$")

Un archivo de comandos puede pedir la ejecución de otro archivo de comandos. El sistema permite hacer esto hasta tres niveles.

Los archivos de comandos son especialmente útiles para ahorrar tiempo a aquellas personas que necesiten llevar a cabo procesos rutinarios o periódicos, ya que, al utilizar un archivo de comandos, no tienen que repetir la misma secuencia de instrucciones cada vez que realicen uno de estos procesos.

1.3 Variables.

La información que se maneja en SAI es principalmente de tipo numérico. Para esto el investigador genera variables que contengan los datos con los que quiere trabajar. Las variables pueden ser de cuatro tipos, escalares, vectores, matrices y series de tiempo. Tomando como serie de tiempo, un vector de datos al que se le asocia una periodicidad y una fecha de inicio.

A cada variable se le asigna un nombre para diferenciarla de las demás. El nombre de una variable puede tener hasta ocho letras o números, siendo obligación que empiece por una letra.

Las variables sirven para generar otras variables por medio de expresiones aritméticas y para alimentar diferentes aplicaciones como gráficas, regresiones o simulación de modelos.

Algunas instrucciones de SAI permiten el manejo simultáneo de un conjunto de variables. Este conjunto se define por medio de características de los nombres de las variables. Por ejemplo, una persona puede solicitar un listado de todas las variables que empiecen con las tres primeras letras "MEX". Por esto es aconsejable que los usuarios utilicen una nomenclatura que relacione el tipo de información que contienen las variables con los nombres de éstas.

1.4 Bancos de Información.

1.4.1 Bancos de Datos.

El investigador puede almacenar sus variables en un banco de datos. En SAI los bancos de datos se crean como archivos individuales e independientes. Cada archivo es como un cajón donde el usuario deposita sus variables.

Las variables en un banco de datos pueden ser creadas, borradas u ordenadas en orden alfabético. También es posible copiar variables de un banco de datos a otro. Además, existe un comando en SAI que sirve para desplegar una lista de un subconjunto de las variables de un banco de datos. Este comando es "CATALOGO".

Cada usuario del sistema tiene definidos varios bancos de datos. De este conjunto de bancos, puede designar opcionalmente uno de ellos como área de trabajo. Para el sistema esto quiere decir que las variables con las que trabaja pertenecen a este banco, a menos de que se especifique lo contrario. Sólo es posible definir un banco a la vez como área de trabajo. Un banco es área de trabajo hasta el momento en que se define otro archivo como área.

Al entrar al sistema, SAI crea y asigna un banco de datos temporal como área de trabajo. En esta área temporal, el usuario puede guardar variables, pero al abandonar el sistema, tanto el archivo temporal como las variables desaparecen.

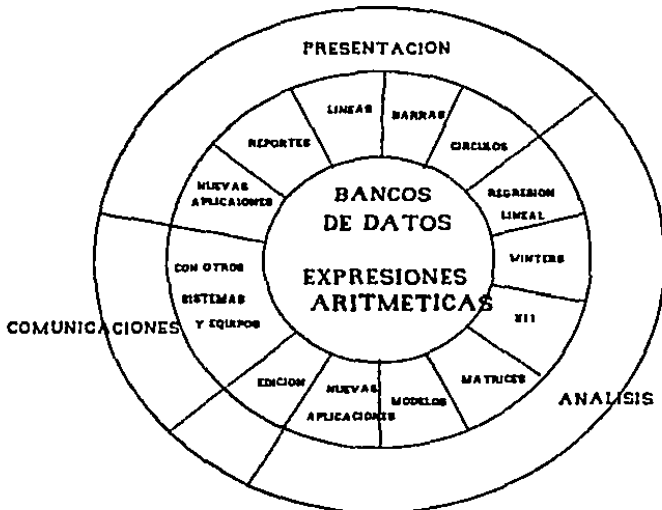
1.4.2 Bancos de documentación.

Opcionalmente los usuarios tienen la posibilidad de crear un banco que contenga información escrita asociada a las variables de un banco de datos. Un banco así, se llama Banco de Documentación. Para cada banco de documentación siempre tiene que existir un banco de datos asociado. El usuario asigna, según sus necesidades, calificativos en el banco de documentación a las variables de un banco de datos. Estos calificativos son dados por el usuario o por el sistema en caso de omisión y se definen al momento de crear el banco de

documentación. Los calificativos son iguales para todas las variables, pero los valores que toman pueden ser distintos para cada una.

1.5 Estructura del Sistema.

SAI esta formado por trece programas que corren en forma independiente y que se comunican entre si. Cada uno de estos programas constituye un módulo de SAI que tiene diferentes funciones y aplicaciones. A continuación se presenta un diagrama que esquematiza la estructura de SAI.



En este diagrama cada región corresponde a uno o más módulos del sistema, los cuales veremos más adelante.

Al entrar a SAI, el analista tiene acceso inicialmente al módulo central del sistema. Ahí, puede realizar diferentes procesos con variables, bancos de datos y bancos de documentación. El usuario transfiere el control a cualquiera de los otros módulos desde el módulo central, regresando a éste una vez que termina.

Cuando un usuario transfiere el control de un módulo de SAI a otro, lo que sucede es que el programa del módulo de salida manda correr el programa del módulo de llegada, parándose una vez que lo haya hecho.

Cuando un usuario transfiere el control a algún módulo, el sistema mantiene vigente el Área de trabajo definida. Esto permite alimentar, con las variables del Área, a los diferentes módulos de SAI.

También, es posible utilizar, en cualquiera de los módulos, variables de bancos diferentes al del Área de trabajo, pero en este caso, el usuario debe especificar a qué banco pertenecen las variables.

Asimismo, el sistema cuando genera variables como resultado de algún proceso llevado a cabo en un módulo, las mantiene en el Área de trabajo y pueden ser utilizadas por otros módulos posteriormente. Esto último es una de las grandes ventajas que tiene un analista al utilizar SAI.

La estructura en módulos de SAI, permite que el sistema pueda crecer casi sin límites. Para crear una nueva aplicación basta con hacer un programa que sea incluido como un nuevo módulo.

1.5.1 Módulos de SAI.

Al entrar el usuario a SAI, el sistema lo posiciona automáticamente en su módulo central. Ahí, el usuario puede generar variables por medio de expresiones aritméticas, obtener catálogos (lista de variables) de los diferentes bancos de datos, crear y manejar los bancos de datos, documentar las variables y generar listados del contenido de las variables. Los catálogos y los listados de variables pueden incluir su documentación.

Desde este módulo, como ya dijimos, el usuario transfiere el control a los demás, regresando a éste una vez que termina. Para trasladar el control a otro módulo, el usuario lo hace usando el comando EJECUTA, al cual tiene que darle como argumento el nombre del módulo que desee. Así, para pasar al módulo de líneas, el analista debe de dar la siguiente instrucción:

EJECUTA LINEAS

Para regresar de cualquiera de los módulos al módulo central, el usuario usa la instrucción FIN. Esta instrucción es de las pocas que no requieren el signo de pesos ("\$") para terminarla. Para abandonar el sistema, el usuario se vale de este comando en el módulo central.

El módulo de EDICION y GENERACION de variables sirve para capturar y editar la información que contienen las variables. Ahí, el usuario puede crear nuevas variables, además de corregir, insertar y eliminar elementos de las variables ya existentes.

Para generar representaciones gráficas de la información contenida en las variables, existen en SAI tres módulos. El módulo de LINEAS, en el cual las gráficas se hacen con líneas continuas. Sirve por ejemplo para mostrar las variaciones de los valores de una variable, su tendencia general y las fluctuaciones en torno a esa tendencia. Para realizar las diferencias en valor de un grupo de variables homogéneas, SAI cuenta con el módulo de BARRAS. En este módulo el usuario puede graficar utilizando, además de las barras comunes, diez diferentes figuras, como la figura de una casa o de un signo de pesos ("\$") entre otras. El módulo de CIRCULOS grafica las variables en forma de círculos divididos, indicando las proporciones de cada dato de una variable con respecto a la suma de todos sus datos. Los tres módulos de graficado de SAI ofrecen al usuario toda una gama de posibilidades para el diseño de una gráfica. Una gráfica puede tener títulos y subtítulos además de otros letreros explicativos de la gráfica. También, el usuario puede especificar escalas, rotulados de los ejes y tipos de líneas que desee. Sin embargo, si el usuario no especifica alguna de estas características, el sistema las calcula automáticamente.

SAI tiene también un módulo para generar cuadros numéricos con los datos de las variables. Se trata del módulo de REPORTES. Este módulo da mayor claridad y presentación, en la impresión de los datos, que los simples listados generados en el módulo central. Las alternativas de diseño de los reportes son varias. El reporte puede tener un título y un subtítulo, letreros en cada renglón y en cada columna y un pie de página, además de la posibilidad de presentar la información en forma vertical u horizontal, entre otras cosas.

Para regresión lineal múltiple SAI cuenta con el módulo de regresiones. Este módulo proporciona diferentes estadísticos, entre otros: Matriz de correlación, matriz de varianzas-covarianza, r cuadrada, Durbin-Watson, análisis de varianzas, estimación de los parámetros y sus intervalos de confianza, prueba T para los estimadores, residuales y componentes principales de las variables independientes, entre otros. El sistema puede corregir la presencia de autocorrelación por el método de Cochran-Orcutt. Existe también, la posibilidad de mantener algunos resultados en variables, como por ejemplo la matriz de correlación y los residuales.

El analista realiza operaciones matriciales en el módulo de MATRICES. Ahí, puede multiplicar matrices y matrices con vectores, transponer e invertir matrices, además de sumar y

restar matrices de la misma dimensión. Así mismo, en este módulo, el usuario puede utilizar el método RAS utilizado en la estimación de matrices de insumo-producto.

Para el manejo de series de tiempo, SAI cuenta con el módulo de WINTERS, el cual pronostica una serie de tiempo através de su descomposición en los componentes: Ciclo, tendencia, ciclo estacional y ruido.

Para el ajuste estacional de series de tiempo, SAI cuenta con el módulo de XII, que utiliza el método CENSUS II versión XI, que descompone las series en los componentes arriba mencionados. Existe un módulo de conexión con IBM que permite utilizar la información guardada en los bancos de datos de SAI en programas de TSP (Time Series Processor), paquete que corre en la computadora IBM-370 de la Dirección de Planeación Hacendaria de la SHCP, con la que existe una conexión por medio de línea telefónica con la PDP 11/34.

Para la simulación de modelos matemáticos, SAI cuenta con dos módulos. Uno de estos, MODELOS, resuelve sistemas de ecuaciones no-lineales en el tiempo. El otro, el módulo de simulación MONTE CARLO, simula modelos estocásticos no simultáneos. Este último módulo es el tema del que trata de este trabajo.

1.4 Expresiones Aritméticas.

El usuario de SAI, define expresiones aritméticas que actúan sobre las variables almacenadas en los bancos de datos. Combinando, con ciertas reglas lógicas, constantes, variables, operadores y funciones, el usuario forma explícitamente la expresión aritmética que desee. SAI provee para este fin, una biblioteca de funciones que no sólo llevan a cabo operaciones matemáticas básicas, sino que efectúan procesos complejos sobre series de tiempo, vectores y matrices.

1.4.1 Operadores.

Existen dos tipos de operadores en SAI, los operadores aritméticos y los operadores lógicos. Al operar dos variables, los primeros dan resultados numéricos y los segundos dan como resultado un cero o un uno, dependiendo de la veracidad de la proposición formada por el operador y los operandos. Un ejemplo de operación aritmética es $5 + 3$, y un ejemplo de operación lógica es $6 > 9$.

Los operadores aritméticos son:

- + Suma
- Resta
- * Multiplicación
- / División

DIV División entera
MOD Módulo (residuo de la división entera)
^ Potencia

Los operadores lógicos son:

Y Conjunción lógica
O Disjunción lógica
MAY Mayor
MEN Menor
MAI Mayor o igual
MEI Menor o igual
IGU Igual
DIF Diferente

La operación entre variables de mismo tipo es una variable de ese mismo tipo, la cual es resultado de la operación independiente y elemento a elemento de los operandos. Por ejemplo, la suma de dos vectores es un vector formado por la suma elemento a elemento de los dos vectores operandos. En el caso de vectores y matrices, es necesario que los dos operandos tengan el mismo número de elementos y en caso de matrices, en mismo número de renglones y columnas. En cuanto a series de tiempo, el sistema hace la operación únicamente en la intersección en el tiempo de las dos series, resultando así, una serie de esa longitud.

Cuando un escalar es operado con un vector, serie de tiempo o matriz, el resultado es respectivamente un vector, una serie de tiempo o una matriz cuyos elementos fueron operados cada uno con el escalar.

En el caso de la multiplicación de dos matrices, no se trata de la multiplicación matricial, sino de una multiplicación elemento a elemento. La multiplicación matricial se calcula por medio de una función llamada MMULT o bien en el módulo de matrices.

1.6.2 Funciones.

SAI cuenta con una biblioteca de funciones ya construidas que se encuentran al alcance del usuario. Para utilizarlas en una expresión aritmética, debe especificar la función seguida de sus argumentos entre paréntesis y separados por comas.

Las funciones en SAI se dividen en diferentes grupos, dependiendo del tipo de aplicación que tienen. Estos grupos son: Matemáticas, de Selección, de Dimensionamiento, Estadísticas y Financieras.

Las Funciones Matemáticas son las más utilizadas comúnmente en matemáticas, tales como valor absoluto, seno, coseno y logaritmos entre otras. Estas funciones con la excepción de MMULT son funciones monádicas (un solo argumento) y se ejecutan para cada elemento del argumento. Por ejemplo, el seno de un vector de dimensión tres cuyos elementos son 4, 2 y 3 es un vector de dimensión tres cuyos elementos son seno de 4, seno de 2 y seno de 3.

Para extraer ciertos elementos de una variable, por ejemplo, el mínimo de un vector, o una parte de una serie de tiempo, los usuarios utilizan las Funciones de Selección de SAI. El usuario También puede referirse a un elemento de una variable por medio de índices. Por ejemplo si A es un vector, entonces A(3) es su tercer elemento. En el caso de matrices es necesario especificar dos índices para referirse a un elemento, pero al usar un sólo índice el resultado es un vector formado por el renglón correspondiente de la matriz. Así si M es una matriz, entonces M(2) es un vector formado del segundo renglón de M. La función COLU sirve para obtener una columna de una matriz.

El analista puede generar variables que contengan ceros por medio de las Funciones de Dimensionamiento. En algunos casos esto le puede ahorrar mucho tiempo y trabajo. Por ejemplo, existe una función que genera una matriz de ceros de ciertas dimensiones. Dentro de esta categoría entra también, la función MATI que genera una matriz identidad y la función ALEAT que genera un vector de números aleatorios con distribución uniforme.

De las funciones más importantes de SAI son las estadísticas. Estas calculan diferentes estadísticos como son: media, varianza, autocorrelación, tasas de crecimiento y flujos (primeras diferencias) de vectores y series de tiempo. Asimismo, SAI tiene funciones para calcular covarianza, correlación y correlación cruzada de dos variables. También, unas de estas funciones agregan y desagregan series de tiempo para convertirlas a series con distintas periodicidades.

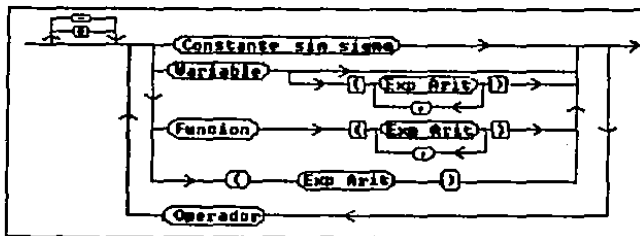
Las Funciones Financieras de SAI calculan tasas de interés, tasas de descuento, valores futuros, anualidades y tablas de amortización.

A las expresiones aritméticas que matemáticamente no están definidas, como por ejemplo la división entre cero o el logaritmo de un número negativo, SAI les asigna el valor de N.D. (valor no definido). Este valor es un número tal que operado con otro número o usado en cualquier función, da como resultado el mismo valor N.D. . Este valor sirve también para el caso en que al capturar una variable, el analista desconozca el valor de uno o más de sus elementos.

En este caso, tiene que especificar el valor no definido, simplemente escribiendo N.D. en las posiciones de los datos desconocidos.

Los resultados de las expresiones aritméticas sirven para definir nuevas variables, sustituir los valores de las ya existentes y para alimentar aplicaciones específicas. El último caso se incluye en todos los módulos de SAI. Por ejemplo podemos pedir un desplegado en pantalla de una expresión aritmética en el módulo central o graficar otra en algún módulo de graficado, todo esto sin crear variables que sean el resultado de estas expresiones.

Una expresión aritmética puede contener a otras expresiones aritméticas, por ejemplo la expresión $A*(5+2)$ contiene la expresión aritmética $5+2$. Una estructura así se dice que es recursiva. La estructura recursiva de la sintaxis de una expresión aritmética en SAI está representada en el siguiente diagrama:



Donde EXP ARIT es una expresión aritmética. En este diagrama se puede ver que las expresiones aritméticas pueden ser usadas al aplicar funciones o al indexar variables.

1.7 Auxilio del Sistema al Usuario.

1.7.1 Ayudas.

En algún momento en una sesión de trabajo en SAI, un usuario puede llegar a necesitar ayuda sobre la sintaxis o función de alguno de sus módulos. Para estas personas, existe en SAI el comando de AYUDA. Al proporcionar este comando seguido del signo "\$" el sistema despliega en pantalla una lista de los diferentes comandos y atributos del módulo en que se encuentre el usuario y una pequeña explicación de ellos. Para obtener información más amplia sobre alguno de los

comandos o atributos, el usuario debe proporcionar el comando de AYUDA seguido del nombre del comando o atributo en particular. En el módulo central es posible también pedir ayuda sobre la forma de utilizar las funciones aritméticas.

1.7.2 Forma conversacional.

Cuando un usuario escribe una instrucción de SAI en más de un renglón, al cambiar de renglón el sistema solicita al usuario el siguiente elemento de información que requiere para su proceso. Esto puede ayudar en un momento dado a un usuario que haya olvidado el tipo de elemento que requiere una instrucción. A esto es a lo que llamamos forma conversacional de dar una instrucción.

Por ejemplo, en el módulo de EDICION la siguiente instrucción genera una serie trimestral con inicio en el segundo trimestre de 1982, con nombre MEX82 y que tiene cinco elementos:

```
EDICION > GENERA MEX82,SERIE,5,4,8202,231,535,342,432,654*
```

(La instrucción empieza a partir de la palabra GENERA, el letrero EDICION es puesto por el sistema). Esta instrucción se puede proporcionar de la siguiente forma:

```
EDICION > GENERA  
VARIABLE> MEX82  
, > ,  
TIPO > SERIE  
# ELEMEN> 5  
, > ,  
PERIODO > 4,  
INICIO > 8202,  
OBS # 1> 231,535  
, >,342,  
OBS # 4>432,654 *
```

(Nótese que en lugar del letrero EDICION, el sistema indica el siguiente elemento de información).

1.7.3 Control de Errores.

El sistema interpreta todos los atributos y argumentos de una instrucción, antes de realizar cualquier procesamiento. En caso de error, antes o después del proceso, el sistema envía un mensaje de error, explicando al usuario el error

cometido, indicándole el punto exacto donde detecto el mismo. El usuario tendrá que repetir la instrucción para que el sistema la ejecute correctamente.

1.8 Referencias.

Para mayor información de las funciones y el manejo de SAI, el lector puede referirse al manual del usuario de SAI [30].

Si el lector tiene interés de conocer la filosofía de programación con la que se realizó el sistema, puede referirse a los dos tomos de Programación lógica de J.D. Warnier [43,44], y a las tesis profesionales de Juan Carlos García [11] y la de Barraza, Blando y Gómez [3].

CAPITULO II El Método de Simulación Monte Carlo.

2.1 Modelos.

La experimentación, en toda ciencia, ayuda en mucho a comprender y controlar los sistemas de la vida real. Es por esto que la experimentación forma una parte esencial de la ciencia moderna. Ella permite a los científicos comprobar y demostrar sus teorías sobre el comportamiento de nuestro universo.

Sin embargo, muchos sistemas, con los que tratan los investigadores, son tan grandes que la experimentación con ellos en un laboratorio representa un costo demasiado elevado o simplemente es imposible de realizar. Por ejemplo, en una guerra, la experimentación con una táctica militar no utilizada anteriormente, en el campo de batalla, tendría un costo en vidas humanas y en posiciones estratégicas de valor incalculable, en caso de no ser eficaz. O también, por ejemplo, imagínese el lector el costo social y económico que tendría una mala estrategia económica, llevada a cabo con el fin de experimentar con ella y sin saber cuales pueden ser sus repercusiones.

Es por esto que el investigador necesita encontrar un método alternativo para experimentar. En el libro "Un Concepto de Planeación de Empresas" Russell L. Akoff [1], dice que para que la experimentación tenga algún sentido y produzca un conocimiento, el científico debe realizarla sobre algo que tenga un comportamiento similar al sistema que está estudiando. Akoff propone la utilización de un modelo que refleje en su funcionamiento las relaciones internas del sistema bajo estudio, pero que tenga una estructura más sencilla.

Según Akoff, un modelo es una representación de un sistema. Puede ser una representación física, como es el caso de una maqueta o de un modelo en miniatura de un coche o avión, una representación gráfica, como es una curva de demanda o también una representación simbólica, como los modelos matemáticos. Sin embargo, en este trabajo sólo nos ocuparemos de los modelos matemáticos.

El estudio de muchos sistemas se hace por medio de modelos matemáticos, los cuales son representaciones simbólicas de ellos. Las ecuaciones, inecuaciones y relaciones lógicas de un modelo matemático deben de reflejar las interrelaciones y restricciones de las variables que miden las características del sistema.

H.P. Williams, en su libro "Model Building in Mathematical Programming" [17], dice que es importante notar que un modelo está definido enteramente por sus relaciones matemáticas. Estas relaciones son independientes de los datos del modelo. Un modelo puede ser usado por un tomador de decisiones en muy

diversas ocasiones con diferentes datos como costos, coeficientes técnicos, recursos disponibles, etc. El objetivo principal de un modelo matemático es la experimentación. No obstante, dice Fishman en su libro "Concepts and Methods in Discrete Event Digital Simulation" [10], la utilización de un modelo matemático tiene otras ventajas, como son:

- 1) Organizar las creencias teóricas y observaciones empíricas que el investigador tiene sobre un sistema.
- 2) Induce a mejorar el entendimiento del sistema
- 3) Pone en perspectiva la necesidad del detalle.
- 4) Facilita el análisis.
- 5) Provee una estructura para probar el deseo de modificar el sistema.
- 6) Es más fácil de manipular que el sistema mismo.
- 7) Permite controlar el sistema sobre más fuentes de variación que si el estudio fuera hecho sobre el sistema real.
- 8) Casi siempre es menos costoso que el estudio sobre el sistema mismo.

Existen varios criterios para clasificar los modelos matemáticos, para este trabajo usaremos principalmente dos, si varían en el tiempo o no y si tienen elementos estocásticos o carecen de ellos. Se dice que un modelo es estático si su comportamiento no depende del tiempo y por el contrario un modelo es dinámico en el caso en que tenga elementos que varíen con el tiempo. Los modelos, en los cuales todos sus elementos e interrelaciones están dadas por relaciones matemáticas y lógicas exactas, son llamados modelos determinísticos. En oposición un modelo estocástico tiene por lo menos un elemento de carácter aleatorio.

2.2 Simulación.

Akoff en su libro "Un Concepto de Planeación de Empresas" [1], define la simulación como a la experimentación con un modelo. Esta tiene la finalidad de aprender sobre el sistema en estudio. Un ejemplo de simulación es el entrenamiento de un piloto aéreo en un simulador de vuelo. En él, el piloto puede ensayar diferentes procedimientos de vuelo, dada una situación de peligro, para que en alguna ocasión en pleno vuelo que se le presente una situación similar, escoja el procedimiento que le haya dado mejores resultados.

Para Frederick S. Hiller y Gerald J. Liberman, en su libro "Operations Research" [16], la simulación no es más que la realización de experimentos muestrales con el modelo de un

sistema. Estos autores nos explican también, que los experimentos se hacen con el modelo y no en el sistema real ya que lo último sería muy inconveniente, costoso y tardado.

Naylor, Balintfy, Burdick y k. Chu en el libro "Computer Simulation Technique" [27] nos explica que la razón fundamental de la simulación es el deseo irrefrenable del hombre de conocer el futuro. Esta búsqueda y el deseo de predecir el futuro son tan antiguos como la historia humana. En efecto, la simulación permite al tomador de decisiones especular sobre el comportamiento futuro de un sistema, dado que ciertas decisiones sean tomadas por él o por otras personas.

La simulación difiere dependiendo del tipo de modelo sobre el que experimenta el investigador. Por ejemplo, la experimentación, con un modelo en miniatura de un coche en un túnel de viento, es una simulación física. Así, definimos como simulación matemática a la que es llevada a cabo sobre un modelo matemático. A partir de este punto, entenderemos como simulación a la simulación matemática, ya que este trabajo sólo tratará de ésta.

Akoff nos explica también, que en ciertos modelos matemáticos es posible que el científico determine cuál va a ser el resultado de un experimento por medios analíticos o deductivos, por lo que no necesita realizar el experimento. En la mayoría de estos casos el camino analítico es el que debe seguir el investigador, puesto que con él, obtiene resultados más precisos que los que alcanzaría através de la simulación, además ahorra tiempo y recursos. Sin embargo, muchos problemas de la realidad son demasiado complejos para ser resueltos por medios analíticos. En estos casos la simulación es un camino viable.

Una de las ventajas de la simulación en computadora consiste en que en ella, la construcción de diversos escenarios es relativamente sencilla. El escenario de un modelo está formado por el conjunto de parámetros y variables exógenas de éste. El analista puede crear escenarios que sean muy optimistas o por el contrario muy pesimistas, con la finalidad de evaluar el modelo en cualquiera de los casos. Por ejemplo en un modelo de petróleo es importante que un tomador de decisiones planteé distintas situaciones en el precio del crudo. Debe de tomar en cuenta precios altos, medianos y bajos para poder ver como afecta esta variable en la economía de su empresa o país. Gracias a la computadora, el investigador realiza un buen análisis de sensibilidad, el cual lo lleva a cabo cuando modifica ciertas variables o parámetros, dejando las demás constantes, para ver como intervienen en el comportamiento del modelo.

Anteriormente, la simulación había sido vista por los analistas como el último recurso de investigación, el que se empleaba cuando todo lo demás había fallado. La simulación era una forma lenta y costosa de analizar un problema que requería una gran cantidad de tiempo y recursos para ser llevada a cabo. Además, simular es un método impreciso, que sólo genera estimaciones estadísticas en vez de resultados exactos, y solamente permite a los analistas comparar diferentes soluciones a un problema. No obstante, gracias a los recientes avances en las metodologías de simulación y computación, la simulación es ahora una de las técnicas más usadas para el estudio de sistemas, como es el caso de algunos sistemas financieros y económicos. Con el advenimiento de las computadoras de alta velocidad y bajo costo, la técnica de simulación de un modelo matemático en computadora es cada vez más utilizada en muchos campos de la investigación. Ejemplos de estos campos son: Economía, Ingeniería, Investigación de Operaciones, etc.

Sin duda, la simulación es una herramienta versátil de un gran valor para los investigadores. Sin ella, una gran cantidad de problemas quedarían sin ser resueltos.

2.3 Método Monte Carlo.

En el libro "Simulation and the Monte Carlo Method" de Reuven Y. Rubinstein [34], se explica que el término Monte Carlo fue introducido, por Von Neumann y Ulam, en la segunda guerra mundial, como un nombre en clave que designaba un proyecto secreto que se llevó a cabo en Los Alamos estado de Nuevo México en los Estados Unidos. Este proyecto tuvo por objetivo el desarrollo de la primera bomba atómica. El método Monte Carlo fue usado en ese entonces para la simulación de la difusión aleatoria de neutrones lentos en material fisionable. Alfred H. Hausrath en su libro "Venture Simulation in War, Business and Politics" [14], describe como el Dr. Gamow trabajó en Nuevo México, aplicando este método a problemas asociados con la bomba atómica. El término hace alusión a la ciudad de Monte Carlo en Mónaco, la cual es famosa por sus casinos en donde los visitantes apuestan en los juegos de azar.

Diferentes autores definen el método de Monte Carlo de distintas maneras, pero esencialmente todas ellas coinciden.

Shan S. Kuo en el libro "Computer Application of Numerical Methods" [39], nos dice que el método de Monte Carlo se usa principalmente para predecir la consecuencia final de una serie de ocurrencias, cada una teniendo su propia probabilidad.

El ingeniero Enzo Molino en "Comunicación e Informática" [25], explica que el método de Monte Carlo consiste en usar números aleatorios para representar incidentes, eventos o hechos concretos.

Samuel S. Shapiro y Alan J. Gross en el libro "Statistical Modeling Techniques" [36], explican que la técnica de simulación conocida como Monte Carlo se basa en un procedimiento que crea información con base en un juego cuyo resultado dependa de factores de suerte. En un estudio de simulación el juego es una representación matemática o funcional y los factores de suerte son variables aleatorias que se usan para representar elementos del sistema.

El método de Monte Carlo consiste en encontrar una solución a un problema, mediante la estimación estadística de parámetros de un sistema aleatorio que es simulado sucesivamente con un modelo en diferentes escenarios. Monte Carlo obtiene estos escenarios de una muestra aleatoria de una población hipotética de escenarios posibles.

Un estudio de simulación, según Shapiro y Gross, envuelve los siguientes pasos:

- 1) Determinar el modelo matemático que relaciona las salidas del sistema con sus elementos o entradas.
- 2) Asignar a cada elemento del modelo una distribución de probabilidad.
- 3) Generar valores aleatorios para cada variable de entrada y usarlos junto con el modelo para calcular los valores de salida.
- 4) Repetir el paso anterior muchas veces para obtener un conjunto de datos representativo de la salida del sistema.
- 5) Resumir el conjunto de datos, infiriendo una distribución empírica y calculando los percentiles y momentos que se deseen.

El primer paso requiere un conocimiento básico del funcionamiento del sistema y debe de ser realizado por especialistas que conozcan muy bien el problema.

Una vez planteado el modelo es necesario, que el investigador seleccione las distribuciones de probabilidad para cada una de las variables exógenas de su modelo. El método para escogerlas varía dependiendo de lo que se conoce de cada una de las variables. Algunos de estos métodos están descritos en los capítulos 3, 4, 5, 6 y 7 del libro de Shapiro y Grass.

El siguiente paso es el de generar valores aleatorios para las distribuciones seleccionadas. Existen varias técnicas para obtener estos valores, las cuales describiremos más adelante en el capítulo III de este trabajo.

Sólo queda reemplazar las variables exógenas del modelo por los valores aleatorios generados. Este proceso se repite muchas veces con la finalidad de obtener los resultados deseados de la simulación.

Los resultados de la simulación con Monte Carlo, son siempre estimaciones de parámetros relevantes para el investigador. Para formar la muestra que sirve para estimar los parámetros relevantes, el investigador tiene que evaluar el modelo un número determinado de veces. En cada una de estas evaluaciones, los valores que arrojados por el modelo se incorporan a la muestra. Los resultados de una evaluación deben ser independientes de los obtenidos por evaluaciones anteriores. El analista determina el número de veces que se evalúa el modelo dependiendo de la confiabilidad con la que desea estimar los parámetros.

2.3.1 El método Monte Carlo para problemas estocásticos.

El investigador se sirve de Monte Carlo para simular modelos estocásticos, en los cuales no conoce el valor exacto de algunas de las variables, pero sin embargo, puede determinar o aproximar sus distribuciones de probabilidad. Expresando el valor de las variables endógenas de la misma forma que los estadísticos que estiman los parámetros de las distribuciones de las variables desconocidas o funciones de estas, el investigador deberá obtener información suficiente para encontrar una solución a su problema.

Las medidas descriptivas más comunes en estadística y que generalmente son los que el investigador busca obtener con el método de Monte Carlo se dividen en dos grupos que son: Medidas de tendencia central y medidas de variabilidad. En el primer grupo tenemos la media, la mediana y la moda, y en el segundo están el rango o recorrido, los cuartiles, los percentiles, la varianza y la desviación estándar. Estas medidas proporcionan al analista una idea de cuál es el comportamiento del sistema en estudio, pero de ninguna forma lo determinan enteramente.

Es importante resaltar aquí, que cuando una variable está en función de variables aleatorias, en general es muy difícil determinar cuál es su distribución de probabilidad a partir de su forma funcional. Existen casos específicos en los cuales se conoce la distribución de probabilidad de una función de variables aleatorias, por ejemplo, sabemos que si

$$X = \sum_{i=1}^k Z_i^2$$

donde Z_i se distribuye independientemente con una distribución $N(0,1)$ para toda $i=1,2,\dots,k$, entonces X se distribuye con una distribución ji-cuadrada con k grados de libertad. Pero en la mayoría de los casos la situación no es tan fácil, por ejemplo para W , definida como

$$W = Y^2 \ln Z$$

donde Y se distribuye $N(\mu, \sigma^2)$ y z como $\exp 1$, no es fácil obtener la forma funcional de su distribución. Monte Carlo puede servir para estos casos.

En el libro de Shapiro y Gross se propone a Monte Carlo como uno de los métodos que sirven para estimar distribuciones de variables endógenas de un modelo cuando éstas están descritas en función de varias variables exógenas que son variables aleatorias.

Ejemplo 1.-

Una compañía maquiladora tiene que comprar maquinaria para producir cierta pieza y tiene la opción de escoger entre dos máquinas con diferentes características. Las principales diferencias que existen entre ellas consisten en:

- 1) El número de piezas que producen en un tiempo dado.
- 2) El porcentaje de piezas defectuosas que producen.
- 3) El costo de producción por unidad en cada una de las máquinas.

Se supone que, en las dos máquinas, el número de piezas producido en un tiempo dado sigue una distribución de Poisson y el número de piezas buenas sigue una Binomial. Tanto el costo por unidad en cada máquina, como el precio que obtiene la compañía por cada pieza, se consideran fijos. El precio de cada pieza es de \$485, el costo de producción por unidad en la máquina 1 es de \$300 y de la máquina 2 es de \$315. Se sabe además que la máquina 1 produce en promedio 16,000 piezas en el intervalo de tiempo dado y en la máquina 2 el promedio es de 15,500. La probabilidad de que una pieza resulte defectuosa en la máquina 1 es de 0.025 y en la máquina 2 es de 0.015.

Definimos las siguientes variables:

U_1 = utilidad en la máquina 1
 U_2 = utilidad en la máquina 2
 N_{b1} = número de piezas buenas producidas en la máquina 1
 N_{b2} = número de piezas buenas producidas en la máquina 2
 N_{t1} = número total de piezas producidas en la máquina 1
 N_{t2} = número total de piezas producidas en la máquina 2

El planteamiento del modelo es el siguiente:

$$U_1 = 485 N_{b1} - 300 N_{t1}$$

donde N_{b1} se distribuye como una $B(N_{t1}, 0.975)$ y N_{t1} se distribuye $P(16,000)$.

$$U_2 = 485 N_{b2} - 315 N_{t2}$$

donde Nb_2 se distribuye como una $B(Nt_2, 0.985)$ y Nt_1 se distribuye $P(15, 300)$.

Para la simulación usamos el módulo de simulación Monte Carlo de SAI y los resultados obtenidos después de correr el modelo 10,000 veces se pueden observar en la siguiente tabla:

	MAQUINA 1		MAQUINA 2	
	MEDIA	DESVIACION	MEDIA	DESVIACION
UTILIDAD	2,765,902	24,395	2,521,942	22,225
TOTAL PIEZAS	15,999	130	15,499	122
PIEZAS BUENAS	15,599	128	15,266	124

En conclusión, es fácil ver que la compañía va a escoger la máquina 1, ya que, a pesar de producir más piezas defectuosas que la máquina 2, es más rentable.

2.3.2

El científico puede usar el método de Monte Carlo para resolver problemas de carácter determinístico, además de los estocásticos. Esto lo consigue al expresar la solución de su problema de la misma forma en la que se expresa un parámetro de algún proceso estocástico. Generando una muestra aleatoria que obtiene al simular el proceso estocástico, el investigador encuentra la solución al problema determinístico estimando el parámetro correspondiente. De esta forma, el analista puede calcular integrales multivariadas, resolver ecuaciones diferenciales y encontrar puntos óptimos de problemas de programación matemática, entre otras cosas.

Es importante decir que comúnmente se busca estimar la Esperanza de alguna variable aleatoria involucrada en el proceso, con el fin de encontrar la solución al problema determinístico.

Históricamente el método de Monte Carlo ha sido usado en diversas aplicaciones. En un principio fué de gran utilidad al examinar la ecuación de Boltzmann. El famoso estadístico Student lo utilizó para estimar el coeficiente de correlación de su distribución t.

Ejemplo 2.-

Sea $g(x) = x \ln(x)$ función definida para todos los reales positivos. Queremos resolver, por el método de Monte Carlo, la siguiente integral:

$$I = \int_1^3 g(x) dx$$

Para lograr esto expresamos I de la siguiente forma:

$$I = \int_1^3 \frac{g(x)f(x)}{f(x)} dx$$

Donde $f(x)$ es la función de densidad de la distribución uniforme en el intervalo $[1,3]$. De esta forma de expresar I deducimos la siguiente expresión

$$I = E \left[\frac{g(x)}{f(x)} \right]$$

lo que implica

$$I = E[g(x)(3-1)]$$

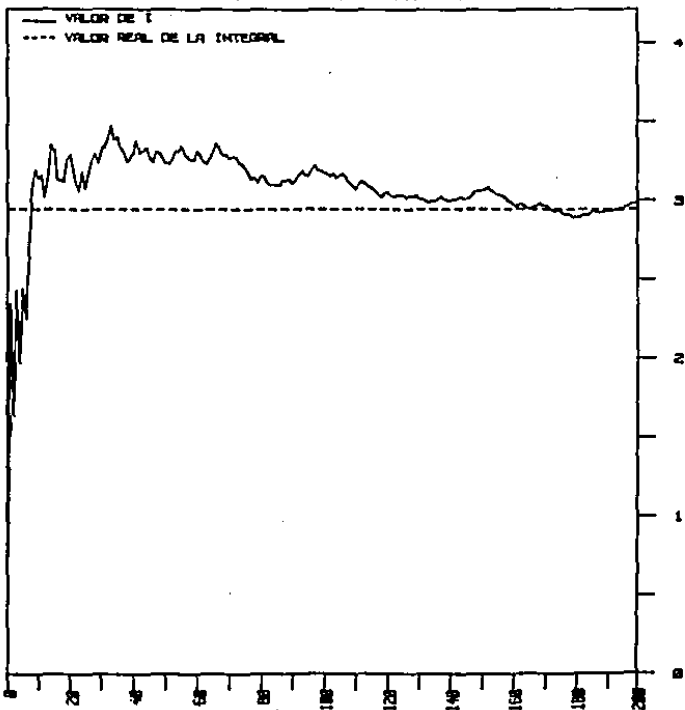
$$I = 2E(g(x))$$

Entonces un estimador insesgado de I es el doble de la media aritmética de $g(x)$, donde x es una muestra aleatoria $U(0,1)$. Para estimar esta última debemos generar un número grande de números aleatorios con densidad $f(x)$, evaluarlos con $g(x)$ y obtener la media de los resultados. Utilizando el módulo de simulación Monte Carlo de SAI, obtenemos que con 100 iteraciones I vale 3.1883, con 1000 vale 3.0614 y con 12000 vale 2.94837. Resolviendo la integral analíticamente sabemos que su valor es aproximadamente de 2.94376, por lo que los errores en cada caso son respectivamente de 0.245, 0.118 y 0.004 aproximadamente.

En la gráfica de la página siguiente se muestra como converge Monte Carlo a la solución en las primeras 200 iteraciones del modelo.

SIMULACION MONTE CARLO
CALCULO DE INTEGRAL

S.R.I.
D.O.S.T.O.



El ejemplo anterior, ilustra como calcular integrales definidas por el método de Monte Carlo. Esto es útil cuando nos encontramos con integrales que no se pueden resolver analíticamente y tenemos que recurrir a métodos numéricos.

CAPITULO III Módulo de Simulación Monte Carlo.

Este capítulo tiene la finalidad de dar una descripción general del módulo de simulación Monte Carlo de SAI, para que el lector comprenda cuales son las herramientas que ofrece SAI al tomador de decisiones que simula con el método de Monte Carlo. El capítulo también da una buena idea del funcionamiento del sistema.

Dentro del contexto de este trabajo, no es necesario que se dé una descripción detallada de la operación y sintaxis del sistema. No obstante, para aquellas personas que les interese y que tengan la posibilidad de utilizar SAI, se incluye al final de esta tesis, el apéndice A que explica la operación del sistema, con todos sus comandos describiendo sus funciones y sintaxis.

3.1 El Módulo de Monte Carlo como parte del Sistema de Análisis interactivo (SAI).

En el capítulo II vimos la importancia que tiene la técnica de simulación Monte Carlo para el estudio de sistemas. Entre los sistemas que los investigadores analizan con esta técnica están los sistemas económicos. Por esta razón, se pensó en la Presidencia de la República, que un módulo para simulación Monte Carlo sería un excelente complemento para el Sistema de Análisis Interactivo desarrollado en esa dependencia.

El módulo de Monte Carlo de SAI tiene por objeto simular modelos matemáticos en los cuales ciertas variables se comportan aleatoriamente, aunque también es posible simular modelos completamente determinísticos. Las distribuciones de probabilidad de estas variables aleatorias deben ser conocidas o supuestas. El sistema es una herramienta suficientemente flexible y general como para simular una gran cantidad de modelos con relativa facilidad.

Es importante notar que la infraestructura de SAI permite complementar al módulo de Monte Carlo con el manejo de sus demás módulos. Para la estimación de parámetros el investigador puede servirse de los otros módulos de análisis. También, el investigador puede valerse del manejo de bancos de datos y expresiones aritméticas para crear los diversos escenarios de su modelo. Finalmente, puede reportar los resultados de la simulación mediante los módulos de graficado y reportes escritos.

El proyecto de Monte Carlo estuvo a cargo del autor y se comenzó a desarrollar en el año de 1983. Para su realización se recibió la asesoría del equipo de trabajo del centro de cómputo de la Presidencia.

Actualmente, el sistema se encuentra en una etapa de desarrollo bastante avanzada. Sin embargo, Monte Carlo es un sistema programado de tal forma que su crecimiento y modificación no implica grandes esfuerzos. Es por esto que el sistema siempre está evolucionando para dar un mejor servicio a sus usuarios.

3.2 Descripción general de los modelos en el módulo de Monte Carlo.

Al construir SAI, existió la preocupación de que el sistema tuviera un manejo de series de tiempo poderoso, por ser ésta una de las estructuras de datos más utilizadas actualmente en la teoría económica. Monte Carlo está enfocado a la simulación de modelos dinámicos, en los cuales las variables principales son series de tiempo.

En el módulo de Monte Carlo de SAI, la simulación puede ser sobre modelos dinámicos. El usuario debe presentar el sistema de ecuaciones en la forma de ecuaciones en diferencia (tiempo discreto). El módulo hace la simulación por recursión hacia adelante (forward recursion). Es decir, en un tiempo t , no se pueden usar resultados de tiempos mayores a t .

Principalmente por razones de tiempo de ejecución y de convergencia, el módulo de Monte Carlo no soluciona sistemas simultáneos de ecuaciones. Las variables endógenas deben estar debidamente despejadas del lado izquierdo de cada ecuación. No obstante SAI permite que las ecuaciones estén dadas en cualquier orden y éste se encarga de ordenarlas para la simulación.

Monte Carlo está diseñado para correr modelos de tipo estocástico, sin embargo, el analista puede simular modelos completamente determinísticos en él. En este caso, el sistema simplemente evaluará el sistema de ecuaciones sin generar números aleatorios.

3.3 Fases del sistema.

El proceso de simulación de Monte Carlo consiste en cuatro subprocesos distintos que en el sistema se denominan Fases. Las Fases de Monte Carlo son procesos que están ligados con la simulación del modelo, pero que el sistema ejecuta independientemente a petición del usuario.

Mediante la Fase Completa, el analista define por primera vez un determinado modelo, o bien redefine uno ya existente. En esta fase el sistema ejecuta la simulación del modelo por primera ocasión.

Posteriormente para llevar a cabo el análisis de sensibilidad, la Fase Datos permite al analista simular un modelo ya definido con diferentes escenarios. Es decir con modificaciones de las variables exógenas y/o los parámetros del modelo.

Por medio de la Fase Resultados, el analista recupera las variables o subconjuntos de ellas que le interesen, después de cada simulación.

El analista, por medio de la Fase Diagnóstico, puede obtener un listado en pantalla o en impresora de ciertos elementos de un modelo que esté definido en Monte Carlo. Elementos como pueden ser el tiempo del modelo, las variables y parámetros que utiliza, entre otras cosas.

En general, el usuario de Monte Carlo no tiene que especificar la fase del proceso que quiere que realice el sistema, sino que el sistema se da cuenta de ello por las instrucciones que ha dado el usuario. No obstante, para ciertos casos, el usuario sí debe de especificar la fase que desea que el sistema lleve a cabo.

3.4 Creación de un Modelo en Monte Carlo.

A continuación veremos los diferentes elementos que constituyen un modelo en el módulo de Monte Carlo, así como los diferentes pasos de la Fase Completa del módulo para la creación de un modelo.

3.4.1 Escenario de un Modelo.

El escenario de un modelo es el conjunto de variables exógenas y parámetros que utiliza un modelo en la simulación. Las variables exógenas deben ser series de tiempo. Los parámetros pueden ser escalares, vectores o matrices. Para crear el escenario, el usuario debe crear un banco de datos en SAI y generar las variables en él. Una vez creado éste, el usuario debe definir el banco de datos como área de trabajo, para que Monte Carlo sepa en dónde obtener las variables. Ya definido el escenario, el usuario puede entrar en el módulo de Monte Carlo para definir su modelo o correr uno ya existente.

Antes de definir un modelo en Monte Carlo el usuario de SAI debe de crear el escenario inicial con el que correrá el modelo. Este escenario será el que utilizará el sistema para la primera simulación. Posteriormente, el usuario en la fase datos puede valerse de otros escenarios para simulaciones subsecuentes.

3.4.2 Nombre de un Modelo.

Lo primero que necesita hacer un analista para crear un modelo en Monte Carlo es proporcionarle un nombre. Este tiene por objetivo diferenciar los posibles modelos que pudiera tener el usuario. El nombre se forma de letras y números, siempre que el primer carácter sea letra, y puede estar formado hasta por seis caracteres.

Inmediatamente después de entrar al módulo de Monte Carlo, es aconsejable, aunque en ciertos casos no es necesario, que el usuario especifique siempre en que modelo va a trabajar. Esto es independientemente de la fase del proceso de simulación que desee realizar.

3.4.3 Tiempo de un Modelo.

Como los modelos de Monte Carlo son en general dinámicos, el usuario tiene que especificar el tiempo con el que quiere simularlos. El tiempo en Monte Carlo está formado por una fecha de inicio, una fecha de finalización y la periodicidad de la simulación. La periodicidad es el número de periodos que forman un año. Una fecha es un número de cuatro cifras en el cual las dos primeras cifras determinan el año y las dos últimas el periodo dentro del año. En el caso de modelos anuales las fechas siempre terminan en 01.

Por ejemplo, si un modelo es trimestral, empieza en el segundo trimestre de 1982 y termina en el tercer trimestre de 1988, entonces su periodicidad es cuatro, su fecha de inicio es 8202 y su fecha de finalización es 8803.

Al definir el tiempo el usuario define también la longitud y la periodicidad de las series endógenas del modelo. Todas las variables exógenas deben de tener la misma periodicidad del modelo, en caso contrario el sistema reporta un error.

En caso de omisión del tiempo, el sistema tomará el intervalo de tiempo y la periodicidad de la variable exógena de menor longitud, en el caso de que exista. En caso de no existir, el sistema reportará un error y pedirá al usuario que especifique el tiempo.

Si un usuario desea simular un modelo que no es dinámico, puede hacerlo simplemente definiendo el tiempo como un solo periodo, es decir, proporcionando la fecha de inicio igual a la fecha de finalización. En este caso, la periodicidad puede ser cualquiera.

3.4.4 Número de Corridas de un modelo.

Los modelos estocásticos tienen que ser evaluados un determinado número de veces para poder estimar los parámetros relevantes del sistema. El número de veces que se evalúa un modelo estocástico en Monte Carlo está determinado por el usuario. Este lo determinará dependiendo de la exactitud que requiera en los resultados.

De no especificar el usuario el número de corridas de su modelo, el sistema supondrá que es uno, por lo que el sistema evaluará el modelo solamente en una ocasión.

3.4.5 Estadísticas de las variables endógenas de un modelo.

El módulo de Monte Carlo permite definir sobre que variables se lleven estadísticos, de modo que después de un número determinado de simulaciones (el cual es especificado por el usuario) , se tenga una estimación de sus medias y desviaciones estandar muestrales.

Los resultados de las estimaciones son guardadas por el sistema en forma de series de tiempo, de manera que en cada periodo de estas series esté la estimación correspondiente. Así, por ejemplo, en el n-ésimo periodo de la serie de medias de una variable X, esta la media de los valores que tuvo X en el n-ésimo periodo, durante todas las corridas de la simulación. Para cada variable para la que el analista desee obtener estos estadísticos, el sistema generará una serie para las medias y otra para las desviaciones estándar. Estas series son recuperadas por el usuario cuando recupera todos los resultados de la simulación y quedan grabadas en el área de trabajo.

Si el usuario quiere especificar cada una de las variables, al las cuales el sistema calcule los estadísticos, debe dar los nombres con los que se creen las variables donde se guarden. Estos tienen que ser diferentes de los de las variables del modelo o de otras variables del área de trabajo, ya que si no es así, el sistema perderá información. Existe la posibilidad de que el usuario pida al sistema que calcule los estadísticos para todas las variables de un modelo. En este caso, el sistema designa automáticamente los nombres de las series de estadísticos, anteponiendo una "m" al nombre de la variable para la serie de las medias y una "d" para el caso de las desviaciones estándar.

Actualmente, el sistema sólo proporciona la media y la desviación estándar, ya que son estos los estadísticos más comunes que se buscan en una simulación. Sin embargo, la programación del sistema está hecha en forma tal que es sumamente sencillo cambiarla para que calcule los estadísticos que requiera algún usuario.

Para obtener el rango de una variable, el usuario puede valerle de las funciones MIN y MAX que se verán más adelante. Definiendo una variable endógena como el valor mínimo que toma la variable y otra para el máximo.

3.4.6 Ecuaciones del modelo.

Como hemos visto, las ecuaciones de un modelo en Monte Carlo son independientes entre sí y en cada una de ellas se define una variable endógena. Las ecuaciones tienen la siguiente estructura: En el lado izquierdo de la ecuación debe de estar el nombre de la variable endógena que se está

definiendo en la ecuación. Del lado derecho tiene que estar la forma funcional de la variable endógena correspondiente. En sí, una ecuación en Monte Carlo es la definición de una variable endógena.

Una variable endógena puede estar en función de variables exógenas, variables endógenas, parámetros, constantes y de retrasos en el tiempo de ella misma o de otras variables. Esta forma funcional tiene la misma estructura que las expresiones aritméticas de SAI (ver capítulo I).

3.4.6.1 Operadores. Los operadores en las ecuaciones de Monte Carlo son los mismos que los operadores matemáticos básicos de las expresiones aritméticas generales de SAI. Estos son:

- + suma
- resta
- * multiplicación
- / división
- ^ exponenciación

No existen operadores lógicos, como tampoco división entera y congruencia módulo.

3.4.6.2 Funciones matemáticas. Las funciones matemáticas para las ecuaciones se utilizan de la misma forma que las de las expresiones aritméticas generales de SAI. Estas funciones son:

SEN	Seno
COS	Coseno
TAN	Tangente
ISEN	Arco seno
ICOS	Arco coseno
ABS	Valor absoluto
EXP	Exponencial
LOG	Logaritmo decimal
LGN	Logaritmo natural
MIN	Mínimo
MAX	Máximo

3.4.6.3 Funciones lógicas. Los operadores lógicos de las expresiones matemáticas de SAI, son remplazados por cuatro funciones con dos argumentos. Estas funciones son:

MAY	Mayor
MAYI	Mayor o igual
IGU	Igual
DIF	Diferente

Estas funciones tienen como resultado un cero o un uno dependiendo de si la proposición que representan es verdadera o falsa. Por ejemplo, MAY(X,Y) es igual a 1 si Y sólo si X es mayor que Y y es igual a 0 en caso contrario.

3.4.6.4 Funciones estocásticas. Para manejar el elemento estocástico, están definidas una serie de funciones que generan muestras aleatorias con cierta distribución de probabilidad. Estas funciones se encuentran al alcance del usuario, quien puede utilizarlas directamente en la definición de las ecuaciones del modelo. Las distribuciones que hasta ahora pueden ser generadas por estas funciones son:

- NORMAL
- UNIFORME Y UNIFORME DISCRETA
- JI-CUADRADA
- EXPONENCIAL
- BETA
- GAMA
- BINOMIAL Y BINOMIAL NEGATIVA
- POISSON
- TRIANGULAR
- GEOMETRICA
- HIPERGEOMETRICA

En el apéndice A de esta tesis, esta una lista de estas funciones, explicando cuáles son los nombres y argumentos que necesitan tener en SAI. Además, hace referencia a los algoritmos que utilizan para generar las sucesiones de números aleatorios y la bibliografía donde se pueden consultar estos.

En el caso de algunas distribuciones de probabilidad, existen funciones alternativas de generación. Cada una de las cuales utiliza un algoritmo distinto. Así, dependiendo de la exactitud y rapidez que desee, el usuario escogerá la función que en particular se apegue más a sus necesidades.

3.4.6.5 Otras funciones. Existen funciones que son exclusivas de Monte Carlo. Estas son la función TABLA, la función ENTREGA y la función ARBOL.

TABLA compara un valor x con una tabla de valores ascendente, estos valores representan intervalos. Dependiendo del intervalo en que se encuentre x , la función consulta una segunda tabla de valores para obtener el resultado. Así, por ejemplo, si tenemos las dos siguiente tablas

$A=(5,8,11,20)$ y $B=(2.3,1.3,4.5,3,7)$

(Nota.- A y B tienen que estar definidos como vectores en el área de trabajo.)

entonces

$TABLA(X,A,B,4)$

(nota.- es necesario especificar la dimensión de las tablas)

vale 2.3 si X es menor que 5, 1.3 si es mayor o igual a 5 pero menor que 8, y así sucesivamente hasta que toma el valor de 7 si X es mayor o igual a 20.

ENTREGA tiene cuatro argumentos que llamaremos x, l, v, p . Esta función compara el valor x con el límite l , si en el período t , x es menor o igual a l , entonces en el período $t+p$ ENTREGA vale v . En todos los demás períodos ENTREGA vale cero. El resultado que toma ENTREGA es el valor que toma v en el período $t+p$. Si antes de llegar a $t+p$ X se vuelve mayor que l , todo el proceso es abortado.

La función ARBOL genera muestras aleatorias de una población dada por un vector de datos, siempre que se le especifique un vector de probabilidades asociado y el tamaño de estos. Por ejemplo, si tenemos los siguientes vectores

$A=(2,3,5)$ y $B=(0.3,0.4,0.3)$

(Nota.- Al igual que en TABLA A y B deben estar definidos en el área de trabajo.)

entonces ARBOL($A,B,3$) valdrá 2 con probabilidad 0.3, 4 con probabilidad 0.4 o 5 con probabilidad 0.3. En caso de que las probabilidades no sumen 1, la función supone que los valores son ponderaciones y por lo tanto, ajusta las probabilidades.

La función ARBOL es una función aleatoria, mientras que TABLA es completamente determinística.

Tanto en TABLA como en ARBOL, los vectores de los argumentos tienen que ser parámetros y por lo tanto, deben de estar definidos en el área de trabajo, antes de la simulación.

3.4.6.6 Ejemplos de ecuaciones escritas para Monte Carlo. Ejemplos de ecuaciones en Monte Carlo son:

```
X = 4.3*Y + 6.7
A = B*C + COS(D+B) / C
PRT = NOR(0,1)
UTILIDAD = VENTAS - COSTOS - GASTOS
MA = MIN(MA,A)
MXA = MAX(MXA,A)
```

Nota.- En las últimas dos ecuaciones se definen el valor máximo y mínimo de la variable A. En este caso, MA y MXA deben de existir anteriormente a la simulación, con un valor en cada periodo. Este valor debe ser tal que permita un buen cálculo de los estadísticos.

El usuario, al ir introduciendo las ecuaciones en el modelo, les asigna un número a cada una. Este número es arbitrario y no impone ningún orden a las ecuaciones, aunque es aconsejable que el usuario vaya asignándolo en el orden en que las introduce. El número tiene la finalidad de distinguir las ecuaciones entre ellas en manipulaciones posteriores.

3.4.7 Variables atrasadas.

Para especificar que una variable está atrasada dentro de la parte funcional de una ecuación, el usuario simplemente pone el nombre de la variable seguido por "(-1)". Por ejemplo, la ecuación

$$X_t = aX_{t-1} + bY_t + cY_{t-1}$$

(donde X es la variable endógena de la ecuación, Y es una variable cualquiera y a,b,c son tres parámetros) debe de escribirse en Monte Carlo de la siguiente forma:

$$X = A*X(-1) + B*Y + C*Y(-1)$$

Por cuestión de capacidad del equipo de cómputo, Monte Carlo sólo permite retrasar una variable un periodo de tiempo. No obstante, es posible que un usuario retrase una variable más de un periodo, utilizando variables auxiliares en más de una ecuación. Por ejemplo

$$X_t = aX_{t-1} + bX_{t-2}$$

en Monte Carlo se logra con las siguientes dos ecuaciones

1) $XR = X(-1)$

2) $X = A*XR + B*XR(-1)$

Anteriormente a la simulación, el escenario deberá contener las variables que estén atrasadas en el modelo, ya sean exógenas o endógenas. Estas variables deberán contener al menos un dato correspondiente al período inmediato anterior a la fecha de inicio del modelo. Este dato es un valor inicial por lo que en si es un valor exógeno al modelo. Por ejemplo, si una de las ecuaciones de un modelo anual que corre entre 7201 y 8601 es

$$X = A \times X(-1) + B$$

entonces antes de simular el modelo, la variable X tiene que existir como una serie anual con por lo menos un dato en el período correspondiente a 7101.

3.4.8 Tiempo de Validez de una Ecuación.

Para facilitar en análisis de sensibilidad, SAI permite que el usuario asigne validez, en un intervalo de tiempo, a algunas de las ecuaciones de un modelo. Esto es especialmente útil para variar el comportamiento de una variable en algún conjunto de períodos.

Estos conjuntos de períodos de validez pueden ser de cuatro formas distintas que son:

- 1) Un fecha.
- 2) Un intervalo de tiempo entre dos fechas (las fechas límite incluidas).
- 3) Un intervalo abierto con una fecha límite dada. Para esta opción existen dos posibilidades, a partir de un fecha o bien hasta cierta fecha (los límites incluidos).
- 4) Cualquier unión de las tres anteriores.

Para lograr la validez de una ecuación en uno de estos conjuntos de períodos, el usuario deberá especificarlo siguiendo la sintaxis que veremos un poco más adelante. El tiempo de validez tiene que ir dentro de la ecuación, entre paréntesis, y entre la variable endógena que se define en la ecuación y el signo de igualdad.

Por fecha entenderemos lo mismo que lo que vimos en el caso de la fecha en el tiempo de un modelo.

Para definir la validez en un período exacto, el usuario simplemente da la fecha correspondiente. Por ejemplo, si en un modelo anual tenemos la ecuación

$$PRM = TRQ + KHM$$

y queremos que sea válida únicamente en el año de 1986 tenemos que escribirla de la siguiente forma:

$$PRM (8601) = TRQ + KHM$$

Cuando el conjunto de validez es un intervalo de tiempo cerrado entonces el usuario debe de dar las dos fechas límites separadas por el signo ":", siendo condición tener que poner la menor en primer lugar. Por ejemplo, si queremos que la ecuación del ejemplo anterior sea válida entre los años de 1982 y 1986 inclusive, entonces debemos de escribir la ecuación así

PRM (B201;B601) = TRG + KHM

Para que una ecuación sea válida hasta cierta fecha, el analista debe poner esta fecha antecedida del mismo signo de ":". Así, si queremos que la misma ecuación de los otros ejemplos sea válida hasta el año de 1979 inclusive, por ejemplo, debemos escribirla de la siguiente manera

PRM (:17901) = TRG + KHM

Por el contrario para que una ecuación sea válida a partir de una fecha, el analista debe de dar la fecha seguida del signo ":". Por lo que, para que nuestra ecuación sea válida a partir del año de 1988, debe de estar escrita de la siguiente manera

PRM (B801:) = TRG + KHM

Para la cuarta opción, el usuario debe de utilizar la misma sintaxis de las opciones anteriores, pero separando cada una por medio de una coma. Supongamos ahora que a nuestra ecuación de los ejemplos anteriores, queremos hacerla válida en la unión de todos los tiempos de validez de los mismos ejemplos, entonces debemos de escribirla de la siguiente forma:

PRM (:7901,B201;B401,B601,B801:) = TRG + KHM

El orden cronológico no es necesario, pero es aconsejable para la comprensión de la ecuación por el usuario. En este ejemplo, la ecuación es válida hasta 1979, entre 1982 y 1984, en 1986 y a partir de 1988.

Es importante que, cuando una ecuación tenga un tiempo de validez, el usuario incluya en su modelo cuando menos una ecuación más, en donde se defina la misma variable endógena en el complemento de tiempo del tiempo de validez de la ecuación. En general, una variable endógena tiene que estar definida para todo el tiempo en que corre el modelo. Cuando una variable endógena no está definida para un periodo, el sistema le asigna un Valor no Definido (ver capítulo I).

Asimismo, los tiempos de validez de las ecuaciones que definen a una misma variable endógena, no deben de tener intersecciones en el tiempo. Para evaluar la variable endógena, el sistema tomará indiferentemente cualquiera de las ecuaciones que definan a la misma variable en un mismo periodo.

Así, el modelo que tiene la ecuación del último ejemplo, tiene que tener también ecuaciones que definan PRM en los años de 1980, 1981, 1985 y 1987 que corresponden al complemento del tiempo de validez de la ecuación. Un ejemplo de una ecuación que cubre este complemento puede ser la siguiente:

$$PRM (8001;8101,8501,8701) = TRG - KHM$$

O por ejemplo, las siguientes dos ecuaciones también lo cubren

$$PRM (8001;8101) = TRG - KHM \text{ y}$$

$$PRM (8501,8701) = TRG / KHM$$

3.4.9 Funciones definidas por el usuario.

El usuario de este módulo de SAI tiene la posibilidad de definir funciones específicas a cada modelo y utilizarlas en las ecuaciones. Estas funciones son particulares a cada modelo y el usuario sólo puede usarlas en el modelo en que estén definidas.

Definir una función es especialmente útil cuando en un modelo se repite varias veces una expresión aritmética, aunque intervengan distintas variables en ella. Al definir una función que remplace a esta expresión, el usuario puede simplificar considerablemente la definición de las ecuaciones de su modelo. Además, el definir funciones tiende a reducir la posibilidad de cometer un error al introducir las ecuaciones y permite la mejor comprensión del modelo.

Para definir una función en un modelo, el usuario debe primeramente designarle un nombre. Este sirve para que el usuario se refiera a la función cuando la utilice en alguna ecuación. Es importante que el usuario verifique que su función no se llame igual que alguna variable o algún parámetro, para que el sistema no los vaya a confundir.

El usuario debe también escoger nombres para cada uno de los argumentos de la función. Estos nombres sirven únicamente para definir la función, pero de ninguna manera deben de corresponder a variables o parámetros del modelo.

La definición de una función está formada por una ecuación. En el lado izquierdo de ésta va el nombre de la función seguido por los nombres de sus argumentos entre paréntesis y separados por comas. En el lado derecho de la misma, debe de ir la forma funcional de ésta. Una función sólo puede tener hasta cinco argumentos.

La forma funcional de una función es simplemente una expresión aritmética de los argumentos de la función y de variables y parámetros del modelo. La sintaxis de estas expresiones es idéntica a la de las ecuaciones. Es decir, los operadores y las funciones son los mismos.

Es posible que para definir una función, el usuario se valga de funciones definidas por él. Sin embargo, es importante que no exista recursividad de ningún orden en la definición de las funciones de un modelo.

Ejemplo de funciones son:

RESTA (G,H) = G - H
NORMA (X,Y) = X^2 + Y^2
COSH(U) = (EXP(U)+EXP(-U))/2

Una vez definida una función en un modelo, el usuario puede hacer uso de ella como si se tratara de alguna de las funciones propias de sistema. Es decir, simplemente escribe el nombre de la función con los argumentos que desee en cada caso, en particular.

Si un usuario, por ejemplo, definió la función RESTA de la misma forma que en el ejemplo anterior, entonces puede escribir la ecuación

$$A = A + B - C$$

de la siguiente forma:

$$A = A + \text{RESTA}(B,C)$$

Al igual que en las ecuaciones, el usuario puede darle validez en el tiempo a la definición de una función. Esto lo logra poniendo el tiempo de validez entre los argumentos de la función y el signo de igual, anteponiéndole una coma. El usuario especifica el tiempo de validez exactamente de la misma forma que en que lo hace para el caso de una ecuación.

Una función deberá tener tantas definiciones como cambios de su comportamiento en el tiempo más uno. Por ejemplo, si definimos la función THC queremos que tenga un cambio de comportamiento en el tiempo, tiene que tener dos definiciones. Así si definimos THC de la siguiente forma:

$$\text{THC}(A,B,C), (:B401) = A*B^C$$

y por otro lado

$$\text{THC}(A,B,C), (B5011) = A*B/C$$

entonces, $\text{THC}(x,y,z)$ será igual a xy^z

hasta 1984 y será igual a $\frac{xy}{z}$

a partir de 1985.

Igual que para las variables endógenas, es importante que las funciones estén definidas para todo el tiempo en que corre el modelo. En el caso de que en un periodo una función no tenga definición, el resultado de evaluarla en cualquier punto será un valor no definido (Ver Valor no Definido Capítulo I).

Es importante también que los tiempos de validez de la definición de una función, no tengan intersecciones de tiempo. En caso de que una función esté definida más de una vez en un periodo, el sistema toma indiferentemente una de las definiciones.

Igualmente, para diferenciar las definiciones de funciones entre ellas, al hacer ciertas manipulaciones como por ejemplo, redefinir alguna de ellas, el usuario les designa un número arbitrario. Asimismo, es aconsejable que el usuario guarde un orden en estos números, con el objeto de comprender mejor el modelo.

3.4.10 Compilación de un modelo en Monte Carlo.

Cuando el modelo ha sido completamente definido por el usuario, este último puede dar la orden de ejecutar la simulación. Esta orden consiste simplemente en teclear el signo de "\$".

Al hacer esto, empezará el proceso de la primera simulación y aparecerá en la pantalla el letrero de

"<COMPILANDO ECUACIONES>"

Este letrero se refiere al hecho de que para hacer más eficiente el proceso computacional, SAI genera un programa en lenguaje FORTRAN para simular el modelo. Este programa es el que se encarga de hacer los cálculos que evalúan el modelo. Podríamos decir que, en cierta forma, Monte Carlo se comporta como un compilador, el cual traduce las especificaciones que el usuario dio para definir el modelo en un programa FORTRAN que lo simula.

En el momento de la llamada compilación, el sistema hace una validación de cada una de las ecuaciones del modelo. De existir un error sintáctico en algunas de las ecuaciones, el sistema lo detecta en ese momento, el proceso de simulación se detiene y el sistema reporta el error cometido, diciendo

al usuario en que ecuación estuvo el error y dándole una pista de su causa, para que pueda corregirlo y volver a dar la orden de ejecución.

En el proceso de compilación, el sistema va interpretando cada uno de los elementos de la ecuación. Al encontrar un elemento que no sea un operador o una constante en la parte funcional de una ecuación, el sistema primeramente chequea si se trata de una variable, ya sea endógena o exógena. En caso de que este elemento no sea una variable, verifica si se trata de un parámetro. Posteriormente, hace lo mismo para las funciones definidas por el usuario y finalmente para las funciones del sistema. Si el sistema no pudo reconocer al elemento como alguna de estas opciones, entonces reportará un error.

El orden, en que el sistema chequea las diferentes opciones al compilar una ecuación, es muy importante. Si por ejemplo, un usuario pone el nombre de una variable igual que el nombre de una función, entonces el sistema reconocerá a éste como la variable y no como la función. Lo mismo pasa con los parámetros y las funciones. De esta manera, al definir una función, el usuario puede cambiar el comportamiento de una función propia del sistema.

Una vez finalizada la compilación de las ecuaciones, de existir funciones definidas por el usuario, el sistema procederá a la compilación de éstas. Al hacer esto, aparecerá en la pantalla el letrero

"<COMPILANDO FUNCIONES>"

La compilación de las funciones es un proceso equivalente al de las ecuaciones.

Para que el programa en FORTRAN pueda evaluar correctamente el modelo, las ecuaciones en él deben de estar en el orden adecuado. Es por esto, que el sistema tiene que llevar a cabo un proceso de ordenación de las ecuaciones. Al realizar el sistema este proceso, aparece en la pantalla del usuario el letrero

"<ORDENANDO ECUACIONES>".

De existir algún tipo de simultaneidad en el modelo, el proceso de ordenación de las ecuaciones detecta este error. Igualmente, el sistema reporta el error, para que el usuario modifique su modelo y pueda ser simulado.

Al terminar la ordenación de las ecuaciones y de no existir ningún problema, el siguiente paso que realiza el sistema es el de desplegar en la pantalla un diagnóstico del modelo. Este es equivalente al que se genera en la Fase Diagnóstico del sistema.

Posteriormente, el programa FORTRAN es a su vez mandado a compilar y a correr automáticamente por el sistema. Para lograr esto, el sistema genera un archivo con instrucciones para el sistema operativo de la computadora. Estas instrucciones invocan al compilador de FORTRAN IV en el caso de la PDP 11 y al de FORTRAN 77 en el de la VAX 780 y mandan a correr el programa ya compilado.

Al correr este programa, la simulación es llevada a cabo, evaluando el modelo el número de veces que haya especificado el usuario. El programa reporta el número de corrida cada cierto número de corridas de la simulación, dependiendo del modelo. Una vez terminada la simulación el control del sistema regresa al módulo de Monte Carlo.

El sistema está diseñado para que trabaje de esta forma, con el objeto de llevar a cabo la simulación más rápidamente que si el sistema evaluará directamente el modelo, con algún tipo de procesador interno de expresiones aritméticas.

A pesar de que el proceso de compilación es relativamente lento, vale la pena, ya que el programa que resulta corre muy rápidamente. Esto es especialmente importante en el caso de la técnica de Monte Carlo, ya que en una simulación es necesario que el sistema evalúe el modelo un número grande de veces, además de que, en general, los procesos de generación de números aleatorios son tardados.

Para las simulaciones subsiguientes del modelo con otros escenarios, el sistema utiliza el mismo programa, por lo que no tiene que volver a realizar el proceso de compilación de nuevo. Esto hace que los análisis de sensibilidad que no afecten la estructura del modelo, sean llevados a cabo muy rápidamente por el sistema.

3.5 Recuperación de los resultados.

Al finalizar una simulación el control regresa al módulo de Monte Carlo. A partir de ese momento el usuario puede recuperar los resultados de la simulación. Estos están constituidos por las variables endógenas del modelo y sus respectivas series de estadísticas.

Antes de dar la orden de recuperar los resultados de una simulación, el usuario puede especificar al sistema que sólo recupere ciertas variables que le interesen. Al hacer esto, el sistema únicamente recuperará las variables especificadas y en caso de existir, sus estadísticas correspondientes. Si el usuario no especifica nada, el sistema recuperará todas las variables del modelo.

La orden de recuperar la información consiste simplemente en el signo de "\$". Esta orden la da el usuario, una vez que especificó el nombre del modelo y la fase, en caso de que no estén especificados de antemano.

Los resultados son grabados en el área de trabajo que esté asignada en el momento de recuperarlos. En caso de que las variables que sean recuperadas ya existan anteriormente en el área de trabajo, el sistema las reemplaza.

Una vez terminada la recuperación, las variables quedan a disposición del usuario, el cual las puede utilizar en cualquiera de los otros módulos de SAI. Así por ejemplo, el usuario puede graficar algunas de las variables de su modelo inmediatamente finalizada la simulación.

Al regresar de una simulación el sistema queda automáticamente en la Fase Resultados, por lo que el usuario ya no tiene que especificarla. Sin embargo, el usuario puede salir de Monte Carlo para recuperar los resultados posteriormente. Para lograr esto último, una vez de regreso en Monte Carlo, el usuario debe de especificar el nombre del modelo y que la fase es la Fase Resultados.

El sistema fué diseñado en esta forma para darle mayor flexibilidad al usuario. Así por ejemplo, puede guardar los resultados en un banco de datos distinto del que contiene el escenario que utilizó para la simulación, esto con la finalidad de no destruir información.

3.6 Fase datos (análisis de sensibilidad).

Cuando un usuario define un modelo en Monte Carlo, el sistema lo mantiene para poderlo utilizar en sesiones posteriores. Un usuario puede simular un modelo el número de veces que quiera, en diferentes sesiones de trabajo.

Volver a simular un modelo ya existente puede tener dos finalidades. Actualizar los resultados de una simulación anterior con el mismo escenario, pero actualizado, o bien realizar un análisis de sensibilidad.

Para simular un modelo ya existente en Monte Carlo, el usuario, de ser necesario, debe especificar la Fase Datos y el nombre del modelo y posteriormente dar la orden de ejecución con el signo de "\$".

Antes de dar la orden de ejecución, el usuario puede especificar sobre que variables o parámetros del escenario hubo modificaciones. En caso de que el usuario no lo especifique, el sistema supondrá que todo el escenario fué modificado. A pesar de no ser una práctica necesaria, es bueno que el usuario haga esta especificación, porque con ella, el sistema toma menos tiempo en realizar la Fase Datos, ya que no tiene que sustituir todas las variables en un archivo intermedio que utiliza.

Al volver a simular un modelo ya existente, Monte Carlo toma el tiempo y el número de corridas de la última simulación, a menos que, el usuario especifique, antes de dar la orden de ejecución, un tiempo o un número de corridas distintos.

Después de que el usuario da la orden de ejecución de la Fase Datos, el sistema manda correr de nuevo el programa del modelo. El programa tomará el escenario nuevo para esta simulación. Al terminar el control regresa a Monte Carlo para que el usuario recupere los nuevos resultados.

Si un usuario desea hacer un cambio estructural a uno de sus modelos, no lo puede hacer por medio de la Fase Datos, sino que es necesario que lleve a cabo la Fase Completa de Nuevo. Esto último tendrá como resultado la creación de un programa nuevo que lleve a cabo la simulación. Por cambio estructural entenderemos al cambio de una o más ecuaciones y/o funciones del modelo, al cambio de periodicidad del modelo, o bien, a la especificación del cálculo de estadísticos de variables que anteriormente no los calculaba el sistema.

3.7 Administración de Ecuaciones y Funciones.

Quando un usuario define un modelo en Monte Carlo, el sistema graba sus ecuaciones y funciones en un archivo que se guarda en disco. Gracias a esto, el usuario tiene acceso a ellas en sesiones subsiguientes para hacer manipulaciones con ellas. Estas consisten en listar, borrar o redefinir algunas de las ecuaciones o funciones.

El usuario se vale de los números que asignó a cada una de las ecuaciones y funciones, para referirse a ellas en estas manipulaciones.

En cualquiera de las fases del sistema, el analista puede listar, en pantalla o impresora, un subconjunto de las ecuaciones o de las funciones de su modelo. Por ejemplo, puede pedir que el sistema liste de la ecuación 3 a la 10, o bien que liste la función 5. Asimismo, puede pedir un listado de todas las ecuaciones o funciones.

Igualmente, el usuario puede borrar subconjuntos de ecuaciones o funciones en la misma forma.

Quando un usuario quiere corregir una ecuación de un modelo, simplemente la redefine en la forma correcta. Para esto, el usuario simplemente define la ecuación como si nunca hubiera existido. Quando un usuario hace esto, el sistema automáticamente borra la ecuación anterior y la reemplaza por la nueva.

3.8 Diagnóstico.

La Fase Diagnóstico de Monte Carlo sirve para generar un listado, ya sea en pantalla o en la impresora, de todos los elementos del modelo.

El sistema genera una lista de las ecuaciones del modelo y de las funciones que fueron definidas por el usuario. Posteriormente da una lista de las variables y de los parámetros. El sistema reporta también, los nombres de las series de estadísticos. Finalmente el sistema reporta la fecha de inicio, la fecha de finalización, el número de periodos simulados y el número de corridas de la simulación.

De no existir uno o más de estos elementos, al listar el diagnóstico, el sistema dice en el reporte que el modelo carece de ellos.

Igualmente, el usuario debe especificar el nombre del modelo y la Fase Diagnóstico, y posteriormente dar la orden de ejecución con el signo de "\$".

El diagnóstico de un modelo puede ser de gran ayuda para el usuario en el caso de que su modelo no se comporte como él esperaba, puesto que le puede dar pistas de cuál puede ser su problema.

A continuación se da un ejemplo de un modelo con el objeto de ver su diagnóstico. Supongamos que un usuario definió en Monte Carlo un modelo con nombre CONTI, anual que empieza en 1982 y termina en 1990, que contiene las siguientes ecuaciones:

```
UTI = VEN - A*COST - B*GAST/COM
COST = LOGN ( 2.3, 4.5)
GAST = LOGN ( 3.1, 6.7)
VEN = NDR (3,9)
```

donde UTI, VEN, *COST* Y *GAST* son las variables endógenas del sistema, *COM* es una variable exógena, A Y B son parámetros, *NDR* es una de las funciones internas de Monte Carlo que genera números aleatorios con distribución normal y LOGN es una función del usuario que definió de la siguiente forma:

```
LOGN (X,Y) = EXP( NDR (X,Y))
```

la cual genera números aleatorios con la distribución Log-normal. Además, el usuario pidió que el modelo corriera 1000 veces y que el sistema sacara estadísticas de todas las variables.

Al pedir un diagnóstico de este modelo, el sistema genera el siguiente reporte:

ECUACIONES:

- 1.- $UTI=VEN-A* COST-B*GAST/COM$
- 2.- $COST=LOGN(2,3,4,5)$
- 3.- $GAST=LOGN(3,1,6,7)$
- 4.- $VEN=NDR(3,9)$

FUNCIONES:

- 1.- $LOGN(X,Y)=EXP(NOR(X,Y))$

DIAGNOSTICO DE MONTE CARLO MODELO: CONTI

VARIABLES

PARAMETROS	UTI	VEN	COST	GAST	COM
------------	-----	-----	------	------	-----

	A	B			
--	---	---	--	--	--

ESTADISTICAS

<u>MEDIA</u>	<u>DESVIACION ESTANDAR</u>
MUTI	DUTI
MVEN	DVEN
MCOST	DCOST
MGAST	DGAST
MCOM	DCOM

PERIODICIDAD: 1

INICIO: 8201

FIN: 9001

NUMERO DE PERIODOS: 9

NUMERO DE CORRIDAS: 1000

3.9 Limites del sistema.

En el sistema, los modelos pueden crecer hasta ciertos limites. Estos están determinados por la capacidad de la máquina en que corre SAI. Los limites son fijos, y el usuario no puede sobrepasarlos, de lo contrario el sistema marca el error y no lleva acabo la simulación.

Se a podido observar en la operación del sistema que para la gran mayoría de los modelos, estos limites nunca son alcanzados. No obstante, el usuario que tenga un modelo que se saiga de los limites, puede dividirlo en dos o más submodelos independientes y simularlos por separado. En caso de que no pueda hacerlo, el usuario deberá buscar algún otro paquete para simular su modelo.

Un modelo puede manejar hasta 150 variables, ya sea endógenas, exógenas o variables de estadísticas. Por cada variable a la que el sistema calcule las estadísticas, el sistema toma en cuenta a dos variables más, una correspondiente a las medias y otra a las desviaciones estándar. Por ejemplo, si un modelo tiene 10 variables, y el usuario pide estadísticas para tres de ellas, entonces el sistema manejará 14 variables.

El límite de parámetros en un modelo es de 100 escalares. Por cada vector o matriz, el sistema toma en cuenta el número de elementos del arreglo. Por ejemplo, si un modelo tiene como parámetros un vector de dimensión 50 y una matriz de 5 por 10, entonces el usuario no podrá incluir ningún parámetro más, puesto que ya llegó al límite.

El número de ecuaciones de un modelo puede ser cualquiera, siempre y cuando el usuario respete el límite de variables que maneje. Este puede ser mayor a 150, puesto que el usuario puede definir una variable endógena en más de una ecuación, en diferentes tiempos de validez.

El usuario puede definir hasta 20 funciones distintas dentro de un modelo. Cada función podrá tener como máximo cinco argumentos.

Cada ecuación y función puede tener hasta 200 caracteres, repartidos en tres renglones como máximo.

Igualmente, el usuario puede pedir estadísticas a por lo máximo 50 variables. Por esto, en el caso de que su modelo maneje más de 50 variables, el usuario no podrá escoger la opción "TODAS" en el atributo ESTADISTICAS.

CAPITULO IV Generadores de Números Aleatorios.

En el capítulo II vimos que el método de Monte Carlo sirve para estimar parámetros de distribuciones desconocidas. Para esto Monte Carlo requiere de números aleatorios distribuidos con las distribuciones seleccionadas para las variables exógenas. El analista que simula con el método de Monte Carlo necesita crear programas o subprogramas en un computador que generen números aleatorios con diversas distribuciones de probabilidad, o bien hacer uso de un sistema ya elaborado que lo pueda hacer.

En este capítulo describiremos los algoritmos más generales con los que se logra la generación de números aleatorios en computadora.

4.1 Generadores de números Aleatorios con distribución uniforme en $[0,1]$.

Actualmente, los lenguajes modernos poseen la aritmética suficiente para generar sucesiones de números pseudoaleatorios. Estos números son principalmente sucesiones de números que estadísticamente tienen las características de variables aleatorias independientes uniformemente distribuidas en el intervalo $[0,1]$.

Según Raul Coss Bu en su libro "Simulación un enfoque práctico" [B], la importancia de los números distribuidos $U(0,1)$ o rectangulares radica en su uso para la generación de variables aleatorias más complicadas que son requeridas en los experimentos de simulación.

Coss Bu sugiere tres formas para obtener los números rectangulares (distribución uniforme): La provisión externa como por ejemplo, las tablas de Rand, generación interna a partir de un proceso físico al azar y la generación interna de sucesiones de dígitos por medio de una relación de recurrencia.

Ciertos autores califican a los números generados por el tercer método como pseudoaleatorios, por ser generados mediante una regla puramente determinística.

Independientemente del proceso que se utilice para la generación de los números rectangulares, nos explica Coss Bu, éstos deben poseer ciertas características deseables que aseguren o aumenten la confiabilidad de los resultados obtenidos de la simulación. Estas características son:

1. Uniformemente distribuidos.
2. Estadísticamente independientes.
3. Reproducibles.
4. Periodo largo (sin repetición dentro de una longitud determinada de la sucesión).

5. Generados a través de un método rápido.

6. Generados a través de un método que no requiera mucha capacidad de almacenamiento de la computadora.

Varias técnicas de generación de números pseudoaleatorios en computadora han sido desarrolladas y probadas. La técnica más utilizada para este fin es la llamada Generador Congruencial mixto que está descrita en libro "Simulation and the Monte Carlo Method" de Reuven Y. Rubinstein [34]. Esta técnica consiste en generar una sucesión de números por medio de una fórmula recursiva basada en el cálculo del residuo de la división euclidiana de una transformación lineal entre un número entero m . Así, se comprueba que si x es un elemento de la sucesión definida por la siguiente fórmula

$$x_{i+1} = (ax_i + c) \pmod{m}$$

donde a y c son números enteros no negativos, entonces la sucesión u definida por

$$u_i = \frac{x_i}{m}$$

a pesar de ser completamente determinística, parece seguir una distribución uniforme en $[0,1]$ y sus elementos son estadísticamente independientes.

El único problema que existe en este método es que el proceso que genera la sucesión de números pseudoaleatorios es periódico con período máximo igual a m , es decir que a partir de un número, no mayor a m , de números pseudoaleatorios generados, la secuencia empieza a repetirse. Este problema se resuelve escogiendo m lo más grande posible para asegurar un número suficientemente grande de elementos distintos en cada ciclo de números y escogiendo adecuadamente los valores de a y c de manera que el período de la sucesión sea igual a m , es decir lo más grande posible. Una sucesión con período igual a m se dice que tiene período completo.

Raúl Coss Bu en su libro "Simulación un enfoque práctico", nos da algunas reglas de selección de m , a y c . Estas son:

a) Selección de m .

Existen las siguientes dos opciones:

1. Seleccionar m de modo que sea el número primo más grande posible, que sea menor que 2^d , donde d es el número de bits que tiene una palabra en la computadora.
2. Seleccionar m como 2.

El valor de a debe ser entero impar, y además no debe ser divisible por 3 ó 5. Además, para que el generador tenga período completo, el valor de a debe de cumplir lo siguiente:

1. $(a - 1) \bmod 4 = 0$ si m es múltiplo de 4
2. $(a - 1) \bmod g = 0$ si g es un factor primo de m

Usualmente "a" es seleccionado como $2^k + 1$ (computador binario) con k mayor o igual a 2.

c) Selección de c .

El valor de c puede ser cualquier constante. Sin embargo, para obtener buenos resultados, el valor debe ser tal que $c \bmod 8$ sea igual a 5 para sistema decimal. Para Rubinstein, c tiene que ser primo relativo de m .

Por lo tanto, para que un programador obtenga buenos resultados con su generador de números aleatorios, debe de escoger a , c y m con respecto a estas reglas. El valor de x_0 , llamado semilla de la sucesión, no tiene mayor efecto sobre las propiedades estadísticas del generador, por lo que el programador lo determina arbitrariamente.

Al generar dos sucesiones de números pseudoaleatorios con la misma semilla se obtienen dos sucesiones idénticas. Esto tiene la ventaja de facilitar al investigador el análisis de sensibilidad, ya que este puede distinguir los cambios en el comportamiento del modelo debidos a las modificaciones en parámetros y variables de los provocados por efectos aleatorios.

El carácter determinístico de la sucesión de números x_i no sesga en nada la simulación, ya que posee las características estadísticas que se necesitan para ello. Por esto, la utilización de números pseudoaleatorios es equivalente a la de números aleatorios. A partir de este punto no haremos diferencia entre ellos.

4.2 Generación de números aleatorios con cierta distribución de probabilidad a partir de una sucesión distribuida uniforme en el intervalo [0,1].

La generación de números aleatorios distribuidos con cierta distribución de probabilidad, se obtiene generalmente utilizando como base, una sucesión de números aleatorios con la distribución uniforme en [0,1]. Manipulando de algún modo una sucesión de números aleatorios con distribución $U[0,1]$, el analista puede generar variables aleatorias con distribuciones como la Normal, la Binomial, la Exponencial, la Gama, etc.

4.2.1 El método de la Transformada Inversa.

El método más utilizado para generar números aleatorios con cierta distribución a partir de una uniforme en $[0,1]$ es el método de la transformada inversa. Este método es general y sirve para obtener sucesiones de números aleatorios con distribuciones tanto discretas como continuas.

El método de la transformada inversa consiste en lo siguiente: Sea X una variable aleatoria con función de distribución acumulada F_x . Como F_x es una función no decreciente, se puede definir su función inversa para cada valor del intervalo $[0,1]$ de la siguiente forma

$$F_x^{-1}(y) = \inf\{x: F(x) \geq y\}$$

para $0 \leq y \leq 1$

Sea U una variable aleatoria con distribución $U(0,1)$, si

$$Y = F_x^{-1}(U)$$

entonces Y tiene distribución acumulada F_x . Demostración:

$$p(Y \leq x) = p[F_x^{-1}(U) \leq x] = p[U \leq F_x(x)] = F_x(x)$$

Por lo tanto, para generar un número aleatorio x con distribución acumulativa F usamos el siguiente algoritmo:

- 1.- Generamos u con distribución $U(0,1)$.
- 2.- Calculamos $x = F^{-1}$

Ejemplo.- La función de densidad de la distribución exponencial está dada por

$$f(x) = \begin{cases} \alpha e^{-\alpha x} & x > 0 \\ 0 & x \leq 0 \end{cases}$$

donde es el parámetro de la distribución. Por lo que su función de distribución acumulada es

$$F(x) = \int_0^x \alpha e^{-\alpha y} dy = 1 - e^{-\alpha x}$$

Lo que implica que la función inversa de $F(x)$ sea

$$F^{-1}(y) = -\frac{\ln(1-y)}{\alpha}$$

Como $1-y$ es también es una variable aleatoria con distribución $U(0,1)$, entonces podemos reemplazar $1-y$ por y sin alterar el objetivo.

Entonces un algoritmo para generar un número aleatorio z con distribución exponencial es:

- 1.- Generar un número aleatorio u con distribución $U(0,1)$
- 2.- Calcular $z = -\frac{\ln u}{\alpha}$

El método de la transformada inversa tiene la desventaja de que para muchas distribuciones de probabilidad no es fácil encontrar su función inversa o ésta no se puede determinar. Por ejemplo, en el caso de la distribución Normal la función de distribución esta dada por

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left[\frac{y-\mu}{\sigma}\right]^2\right) dy$$

donde σ y μ son sus parámetros. La única forma de calcular la función inversa de esta función es por medio de métodos numéricos. Por esto, la generación de números aleatorios con distribución Normal, con el método de la transformada inversa, implica la utilización de mucho recursos como tiempo de máquina.

La técnica de la transformada inversa se utiliza para generar números aleatorios con distribuciones, entre otras, como La Exponencial, la Weibul, la Triangular, la Cauchy y en general todas las distribuciones discretas. Aunque también, existen otros algoritmos con los cuales se puede simular estas distribuciones.

4.3 Métodos Alternativos para generación de números aleatorios a partir de sucesiones de números aleatorios con distinta distribución.

Reuven Y. Rubinstein propone en su libro "Simulation and the Monte Carlo Method" [34] dos métodos para generar números aleatorios a partir de sucesiones de números aleatorios con otras distribuciones de probabilidad. Estos métodos son: El método de composición y el método de aceptación y rechazo.

4.3.1 Método de Composición.

Al usar la técnica de composición el analista debe expresar la función de densidad f que quiere simular como la combinación probabilística de una familia $\mathcal{H}(x; z)$ de funciones de densidad apropiadas. En esta familia $\mathcal{H}(x; z)$, z es un parámetro que identifica una sola función de densidad $h(x)$.

Si el valor de z es obtenido aleatoriamente de una función de distribución acumulada $F_z(z)$ y si x se genera también aleatoriamente a partir de $h(x)$ para esa z escogida, entonces la función de densidad de x es

$$f(x) = \int h(x|z) dF_z(z)$$

o si z es entero

$$f(x) = \sum_i P(z=i) h(x|z=i)$$

Raúl Coss Bu en su libro "Simulación un enfoque práctico" [8], nos explica que mediante este método, la distribución de probabilidad $f(x)$ se expresa como una mezcla de varias distribuciones de probabilidad $f_i(x)$ seleccionadas adecuadamente.

Este autor también nos explica los pasos requeridos para la aplicación de este método que son:

1. Dividir la distribución de probabilidad original en sub-áreas.
2. Definir una distribución de probabilidad para cada sub-área
3. Expresar $f(x)$ en la siguiente forma:

$$f(x) = A_1 f_1(x) + A_2 f_2(x) + \dots + A_n f_n(x)$$

$$\text{con} \\ \sum A_i = 1$$

4. Obtener la distribución acumulada de las áreas.
5. Generar dos números aleatorios A_1, A_2 .
6. Seleccionar la distribución de probabilidad $f_i(x)$ con la cual se va a simular el valor de x . La selección de ésta se obtiene al aplicar el método de la transformada inversa, en el cual el eje Y está representado por la distribución acumulada de las áreas, y el eje X por las distribuciones $f_i(x)$. Para esta selección se utiliza A_1 .
7. Utilizar A_2 en el método de la transformada inversa con la distribución $f_i(x)$, para simular el valor de x .

Esta técnica permite generar distribuciones complicadas a partir de distribuciones generadas por métodos más sencillos.

Ejemplo.- Generar x con la distribución

$$f(x) = \frac{1}{6} + x - 1 \quad 0 \leq x \leq 2$$

escribimos $f(x)$ de la siguiente forma

$$f(x) = \frac{1}{3}f_1(x) + \frac{2}{3}f_2(x) \quad 0 \leq x \leq 2$$

donde

$$f_1(x) = \frac{1}{2}$$

$$y \quad f_2(x) = \frac{3}{2}(x-1)^2 \quad 0 \leq x \leq 2$$

así generamos

$$x = 2u_1 \quad \text{si} \quad u_1 \leq \frac{1}{3}$$

o

$$x = 1 + \sqrt{2u_2 - 1} \quad \text{si} \quad u_1 > \frac{1}{3}$$

donde u_1 y u_2 tienen distribución $U[0,1]$.

4.3.2 Método de Aceptación y Rechazo.

Este método consiste en generar números aleatorios con una distribución de probabilidad apropiada y pasar por una prueba a cada uno para determinar si es aceptable o no. Si podemos expresar $f(x)$ de la siguiente forma

$$f(x) = Ch(x)g(x)$$

donde $c \geq 1$, $h(x)$ es una función de densidad y $0 < g(x) \leq 1$, entonces podemos utilizar este método para generar x tal que $f(x)$ sea su densidad de probabilidad. Generamos dos números aleatorios, U con distribución $U[0,1]$ y V con distribución $h(x)$ y vemos si $U < g(V)$. Si esta desigualdad es verdadera entonces aceptamos V , si por el contrario, es falsa entonces tratamos con otro par U y V .

El método se basa en el hecho de que

$$f(x | U \leq g(V)) = f(x)$$

Demostración: (Por el teorema de Bayes)

$$f_1(x|U \leq g(V)) = \frac{P(U \leq g(V)|V=x)h(x)}{P(U \leq g(V))} \quad (1)$$

calculamos directamente

$$P(U \leq g(V)|V=x) = P(U < g(x)) = g(x)$$

$$P(U < g(x)) = \int P(U < g(V)|V=x)h(x)dx$$

$$= \int g(x)h(x)dx = \int \frac{f(x)}{C} dx = \frac{1}{C}$$

reemplazando en (1) tenemos

$$f(x|U < g(V)) = Cg(x)h(x) = f(x)$$

En el libro "Simulación un enfoque práctico" de Raúl Coss Bu [8], viene explicado el método de rechazo para $f(x)$ distribución de probabilidad acotada y con rango finito, es decir, $a \leq x \leq b$. De acuerdo a esta función de probabilidad, la aplicación del método de rechazo implica el desarrollo de los siguientes pasos:

1. Generar dos números uniformes r_1 y r_2 .
2. Determinar el valor de la variable aleatoria x de acuerdo a la siguiente relación lineal de r_1 :

$$x = a + (b-a)r_1$$

3. Evaluar $f(x)$.
4. determinar si la siguiente desigualdad se cumple:

$$r_2 \leq \frac{f(x)}{M}$$

donde M es la moda de la distribución. Si se cumple esta desigualdad entonces se debe utilizar x . De lo contrario, es necesario pasar nuevamente al paso 1 tantas veces como sea necesario.

Coss Bu nos hace notar que algunos autores como Tocher, han demostrado que el número esperado de intentos para que x sea aceptado es igual a M . Por esto, este método es ineficiente para distribuciones de probabilidad con moda grande.

Con el método de Aceptación y Rechazo se obtienen números aleatorios con distribuciones como Beta, Normal y Gama, entre otras. Ejemplo.- Generar números aleatorios que sigan la siguiente distribución:

$$f(x) = 2x \quad \text{si} \quad 0 \leq x \leq 1$$

O de otra forma

para esta función, $a = 0$, $b = 1$ y $M = 2$. Para generar números aleatorios con esta distribución se deben seguir los siguientes pasos:

1. Generar dos números r_1 y r_2 con distribución $U(0,1)$.
2. Calcular $x = r_1$.
3. Si $r_2 > r_1$, entonces se toma el valor de x . Si $r_2 < r_1$, entonces se requiere regresar al paso 1 tantas veces como sea necesario.

4.3.3 Métodos Particulares.

Además de estos métodos, existen métodos particulares para el caso de ciertas distribuciones de probabilidad. Estos métodos aprovechan alguna propiedad de la distribución que se quiere simular.

Por ejemplo, si queremos generar números aleatorios con distribución Normal con media cero y varianza uno, podemos utilizar el teorema central del límite. Este teorema dice que si X_1, X_2, \dots, X_n son variables aleatorias independientes e idénticamente distribuidas con esperanza μ y varianza σ^2 entonces

$$\left(\frac{\sum_{i=1}^n X_i}{\sqrt{n}} \right) - \frac{n\mu}{\sqrt{n}\sigma}$$

tiende aproximadamente a distribuirse $N(0,1)$. Esta aproximación será más exacta entre más grande sea n . Si tomamos una muestra de doce números aleatorios $\{u_1, u_2, \dots, u_{12}\}$ con distribución $U(0,1)$ y calculamos x de la siguiente forma

$$x = \frac{\left(\sum_{i=1}^{12} u_i \right) - 12}{\sqrt{12}}$$

donde μ es la media de $U(0,1)$ y σ^2 su varianza, x tendrá aproximadamente distribución $N(0,1)$. Tomamos 12 números aleatorios ya que si reemplazamos μ y σ^2 por sus valores, el cálculo de x se simplifica mucho. Así, hacemos μ igual a $1/2$ y σ^2 a $1/12$ tenemos

$$x = \frac{\left(\sum_{i=1}^{12} u_i\right) - \frac{12}{2}}{\sqrt{12}\sqrt{\frac{1}{12}}}$$

$$x = \left(\sum_{i=1}^{12} u_i\right) - 6$$

Entonces, para generar un número aleatorio con distribución $N(0,1)$ se puede seguir el siguiente algoritmo:

- 1.- Generar doce números aleatorios
- 2.- Sumarlos
- 3.- Restarle 6 a la suma

Este método da buenos resultados para muchos casos y es muy sencillo de programar. No obstante, si el analista requiere de resultados más confiables, puede recurrir a cualquiera de los otros métodos.

Casi cualquier distribución de probabilidad se puede simular con un algoritmo que explote alguna característica de ésta. Así, se forman sucesiones de números aleatorios distribuidos como Normal, Exponencial, Beta, Lognormal, Binomial, Poisson, Geométrica, etc.

CAPITULO V Pruebas de Bondad de Ajuste y Aleatoriedad.

Como hemos visto anteriormente, los sistemas que utilizan la técnica de simulación Monte Carlo requieren de generadores de números aleatorios con diversas funciones de distribución de probabilidad. En el capítulo III vimos la metodología para que un analista de sistemas pueda hacer rutinas que generen las diferentes series de números aleatorios. Ya hechas las rutinas, es importante que el analista compruebe estadísticamente que sus generadores sean adecuados.

Un generador es adecuado cuando los números generados por él, pertenecen realmente a la población a la que se supone deben de pertenecer, además de seguir un orden aleatorio. Por ejemplo, si el sistema utiliza al generador de una Normal con parámetros μ y σ , el programador debe primeramente evaluar que tanto los números generados se distribuyen con esta distribución, y posteriormente debe comprobar la aleatoriedad.

Para este propósito la teoría estadística ofrece diferentes pruebas de bondad de ajuste y de aleatoriedad. Entre las pruebas de bondad de ajuste más utilizadas están la prueba de tipo ji-cuadrada y las de tipo Kolmogorov Smirnov. Siendo esta última la que analizaremos en este capítulo. Esta prueba fue la que se utilizó para analizar el comportamiento del Módulo de Monte Carlo de SAI.

En el Módulo de simulación Monte Carlo de SAI, con excepción de algunos casos, los generadores de números aleatorios son rutinas de la librería IMSL (International Mathematical Statistical Libraries, INC) [18].

La librería IMSL es un conjunto muy grande de rutinas para resolver problemas de tipo matemático y estadístico. Estas rutinas están escritas en FORTRAN y utilizan algoritmos de gran exactitud diseñados por expertos. Antes de ser liberada, cada rutina de IMSL es probada exhaustivamente, por lo que el usuario de IMSL está seguro de obtener buenos resultados al usarlas. Comprendidas en este conjunto de rutinas, existen generadores de números aleatorios. Estos últimos, son las que utiliza Monte Carlo para generar las sucesiones de números aleatorios.

La Presidencia de la República contrató el servicio de IMSL por lo que tiene el derecho de utilizar una copia, sin tener que pagar otros derechos.

Dado que los generadores de números aleatorios de IMSL fueron probados por los expertos de IMSL, repetir las pruebas de bondad y aleatoriedad de todas ellas no tendría sentido. Sin embargo, para estar completamente seguro de que el sistema no distorsiona estos resultados, se hicieron las pruebas de bondad de ajuste para todos los generadores de distribuciones continuas utilizadas en Monte Carlo. Se trabajó con estas últimas, puesto que para poblaciones de tipo continuas, la prueba de Kolmogorov Smirnov es

más sencilla de aplicar. Para los dos generadores dentro de Monte Carlo que no forman parte de IMSL y que generan números aleatorios con distribución Normal y Triangular, también fue necesario realizar las pruebas.

A continuación se describe la metodología de prueba utilizada por Kolmogorov Smirnov, así como algunos de los elementos que se utilizan en este proceso.

5.1 Función de Distribución Empírica.

La función de Distribución Empírica de una muestra es un estimador de la Función de Distribución real de la población de donde se tomó la muestra.

Si x_1, x_2, \dots, x_n , es una muestra aleatoria de una población $F(\cdot)$, definimos la función de distribución empírica de la siguiente forma:

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(-\infty, x](X_i)$$

donde n es el tamaño de la muestra y

$$I(-\infty, x](X_i) = \begin{cases} 1 & \text{si } X_i \in (-\infty, x] \\ 0 & \text{si } X_i \notin (-\infty, x] \end{cases}$$

Nota. -

$$\sum_{i=1}^n I(-\infty, x](X_i)$$

representa al número de X_i menores o iguales a x .

En el capítulo VI de libro "Introduction to the Theory of Statistics" de Alexander Mood, Franklin Graybill y Duane Boes [26], se presenta el siguiente teorema:

$$P\left(F_n(x) = \frac{k}{n}\right) = \binom{n}{k} F(x)^k (1-F(x))^{n-k}$$

La demostración es la siguiente:

$$\text{Sea } Z_i = I(-\infty, x](X_i)$$

Entonces Z_i sigue la distribución de Bernoulli con parámetro $F(x)$. Por lo tanto

$$\sum_{i=1}^n Z_i$$

se distribuye binomialmente.

$$\begin{aligned} \text{Así } P\left(\sum_{i=1}^n Z_i = k\right) &= \binom{n}{k} F(x)^k (1-F(x))^{n-k} \\ &= P\left(F_n(x) = \frac{k}{n}\right) \end{aligned}$$

$$\text{ya que } F_n = \frac{1}{n} \sum_{i=1}^n Z_i$$

Del teorema anterior se derivan los siguientes resultados:

$$\begin{aligned} E(F_n(x)) &= E\left(\frac{1}{n} \sum_{i=1}^n Z_i\right) = \frac{1}{n} E\left(\sum_{i=1}^n Z_i\right) \\ &= \frac{nF(x)}{n} = F(x) \end{aligned}$$

$$\begin{aligned} \text{Var}(F_n(x)) &= \text{Var}\left(\frac{1}{n} \sum_{i=1}^n Z_i\right) = \frac{1}{n^2} \text{Var}\left(\sum_{i=1}^n Z_i\right) \\ &= \frac{nF(x)(1-F(x))}{n^2} = \frac{1}{n} (F(x)(1-F(x))) \end{aligned}$$

Las dos ecuaciones anteriores enseñan que para una x fija $F_n(x)$ es un estimador insesgado de $F(x)$ y que además es consistente en el sentido del error cuadrado medio, independientemente de cómo sea la forma de $F(x)$. Esto último, según la sección 3.3 del capítulo VII del libro de Mood [26], implica que

$$P\left[\sup_x |F_n(x) - F(x)| \rightarrow 0\right] = 1$$

La expresión anterior es conocida como el teorema de Glivenko-Cantelli y establece que $F_n(x)$ converge uniformemente a $F(x)$ en x con probabilidad 1. El teorema de Glivenko-Cantelli se puede escribir también de la siguiente forma:

$$P\left(\lim_{n \rightarrow \infty} |F_n(x) - F(x)| = 0\right) = 1$$

Por último, como $F_n(x)$ es la media muestral de las variables aleatorias Z_i , entonces por el teorema central del límite, sabemos que $F_n(x)$ se distribuye asintóticamente como una Normal con media $F(x)$ y varianza

$$\frac{F(x)(1-F(x))}{n}$$

5.2 Prueba de Bondad de Ajuste Kolmogorov-Smirnov.

La prueba de bondad de ajuste de Kolmogorov-Smirnov tiene la finalidad de, dada una muestra aleatoria M de una población desconocida pero supuesta de cierta forma, contrastar la siguiente hipótesis

$$H_0: F_n(x) = F_0(x)$$

donde $F_n(x)$ es la función de distribución empírica de M y $F_0(x)$ es la función hipotética de la población de donde se tomó M .

Sea $M = \{x_1, x_2, \dots, x_n\}$ una muestra aleatoria de una población $f(\cdot)$ desconocida. En la sección anterior vimos que $F_n(x)$ es un buen estimador de $F(x)$ distribución desconocida de la población $f(\cdot)$. Por esto, es bastante lógico comparar la función de distribución empírica $F_n(x)$ con $F_0(x)$ la función hipotética de $f(\cdot)$, para evaluar que tanto difiera $F_n(x)$ de $F_0(x)$.

Una forma de comparar $F_n(x)$ con $F_0(x)$ es tomar la máxima distancia vertical entre las dos curvas generadas por estas funciones. Es decir, evaluar

$$K_n = \sup_x |F_n(x) - F_0(x)|$$

K_n es el estadístico sugerido por Kolmogorov en 1933, el cual mide que tan lejos está $F_n(x)$ de $F_0(x)$.

En el libro de Mood, Graybill y Boes [26] en el capítulo once página 508 se postula es siguiente teorema:

Si $x_1, x_2, \dots, x_n, \dots$ son variables aleatorias independientes e idénticamente distribuidas con función de distribución acumulada $F(x)$ y si

$$D_n = \sup_x |F_n(x) - F(x)|$$

donde $F_n(x)$ es la función de distribución empírica, entonces

$$\begin{aligned} \lim_{n \rightarrow \infty} P(D_n < x) &= \lim_{n \rightarrow \infty} P(\sqrt{n} D_n < x) \\ &= \left(1 - 2 \sum_{j=1}^{\infty} (-1)^{j-1} e^{-2j^2 x^2}\right) I(0, \infty)(x) \\ &= H(x) \end{aligned}$$

Como se puede ver, $H(x)$ no depende de la población de donde la muestra fue obtenida, es decir que F_0 es libre de distribución (fuertemente no paramétrico), cuando n tiende a infinito. Este hecho hace que D_n sea muy usado como prueba de bondad de ajuste.

Sea H_0 la hipótesis de que la distribución de una población muestrada es una distribución específica $F_0(x)$, es decir la hipótesis nula

$$H_0: X_i \sim F_0(x)$$

donde $F_0(x)$ está completamente especificada

Sea

$$K_n = \sqrt{n} \sup \{ |F_n(x) - F_0(x)| \}$$

Si H_0 no es falsa, entonces no existe evidencia para llegar a que K_n se distribuye aproximadamente como $H(x)$. Si es falsa, entonces $F_n(x)$ va tender a estar cerca de $F_0(x)$ la distribución real de X_i por lo tanto

$$\frac{K_n}{\sqrt{n}} = \sup \{ |F_n(x) - F_0(x)| \}$$

tenderá a ser grande. Por todo esto, resulta razonable tomar como criterio, el de rechazar H_0 si

$$\frac{K_n}{\sqrt{n}}$$

es grande.

Como K_n se distribuye aproximadamente como $H(x)$ cuando H_0 es verdadera, se ha tabulado $H(x)$ para algunos valores de x , para poder determinar $k_{1-\alpha}$ de tal forma que

$$1 - H(k_{1-\alpha}) = \alpha$$

y así $P(K_n > k_{1-\alpha}) \sim \alpha$

La prueba definida por: Rechazar H_0 si y sólo si $K_n > k_{1-\alpha}$, es conocida como la prueba de bondad de ajuste de Kolmogorov-Smirnov.

A continuación se presenta una breve tabla de los valores de $k_{1-\alpha}$ con respecto a α , esta tabla está especificada en el libro de Mood, Graybill y Boes [26].

α	.99	.95	.90	.85	.80
$k_{1-\alpha}$	1.63	1.36	1.22	1.14	1.07

Ciertos libros como el de W. J. Conover "Practical Nonparametric Statistics" [7], dan tablas aproximadas para la función de distribución de D_n y no de K_n , en este caso el analista debe calcular el valor de K_n y compararlo con las tablas sin multiplicar previamente por \sqrt{n} .

Existe otras pruebas de bondad de ajuste además de las prueba del tipo Kolmogorov-Smirnov, entre otras la prueba es la prueba de ji-cuadrada. según W. J. Conover, la prueba de Kolmogorov es más potente que la prueba ji-cuadrada para la mayoría de los casos.

5.3 Resultados de las pruebas en los generadores del módulo de Monte Carlo.

Para cada generador, tomé diez muestras de tamaño 100, diez de 500 y diez de 1000, realizando la prueba para cada una. Esto con la finalidad de ver cómo se comportan los generadores para muestras de tamaño chico, mediano y grande.

Al realizar las pruebas, algunas rechazan la hipótesis nula. Si el porcentaje de rechazos es chico, no existe evidencia estadística para concluir que el generador es malo. Estos rechazos se presentan por efectos de la aleatoriedad de los generadores.

5.3.1 Rutina para prueba de bondad de ajuste Kolmogorov-Smirnov de IMSL.

Para llevar a cabo las pruebas de bondad de ajuste para los generadores de SAI, se utilizó la rutina NKS1 de la librería IMSL. Esta rutina sirve para comprobar si una muestra pertenece a una población hipotética, utilizando la metodología Kolmogorov-Smirnov.

NKS1 tiene como entrada un vector con la muestra aleatoria de la población real. La muestra tiene que estar ordenada ascendentemente.

El usuario de esta rutina tiene que proporcionar además, una subrutina en FORTRAN que calcule el valor de la función de distribución hipotética en puntos dados. Esta subrutina tiene que estar declarada "EXTERNAL" en la rutina que manda llamar a NKS1. Al llamarla, NKS1 proporciona un vector de seis posiciones con los resultados de la prueba. Estos resultados son los siguientes:

$$1. \quad K_n^+ = \sup_x \{F_n(x) - F(x)\}$$

$$2. \quad K_n^- = \sup_x \{F(x) - F_n(x)\}$$

3.- $K_n = \sup_x |(F_n(x) - F(x))|$

4.- $\sqrt{n}K_n$

5.- Probabilidad de que el estadístico exceda $\bar{\alpha}_n$, cuando la hipótesis de una cola es verdadera.

6.- Probabilidad de que el estadístico exceda $\bar{\alpha}_n$, cuando la hipótesis de dos cola es verdadera.

para efectos de las pruebas de los generadores de SAI, sólo interesan las opciones 1, 4 y 6.

El manual de IMSL dice: "NKS1 obtiene las probabilidades (5,6) asintóticamente. Para n pequeños. El usuario es advertido en el caso en que n sea menor que 80. Para n pequeños, son consultadas tablas (ref. 1, 2 y 3), usando K_1 , K_2 o K_3 . La probabilidad en 5, es usada para pruebas de una cola, pero el usuario debe saber cuál alternativa le concierne una interpretación acertada. K_1 y K_2 son importantes para esta interpretación."

Referencias: Biernbaum Z. W. "Numerical Tabulation of the Distribution of Kolmogorov's Statistic for the Finite Sample Size" [4], Bradly J. V., "Distribution - Free Statistical Tests, Prentice - Hall" [5], Feller W. "On the Kolmogorov-Smirnov limit theorems for empirical distributions" [9], Gnedenko "The Theory of Probability" [12], Lindgren B. W., "Statistical Theory"[23].

5.3.2 Tablas de resultados.

En las siguientes páginas, se presenta dos tablas con los resultados de las pruebas de bondad de ajuste para los generadores de distribuciones de tipo continuo de SAI. En una, la confiabilidad de las pruebas es del 90% y en la otra del 95%.

**RESULTADOS DE LAS PRUEBAS DE BONDAD DE AJUSTE
PARA LOS GENERADORES DE MONTE CARLO
(Distribuciones Continuas)**

Distribución y generador	Parámetros	Rechazos n=100	Rechazos n=500	Rechazos n=1000	Total	Porcentaje
Uniforme (UNI)	0,1	0	1	2	3	10%
Cauchy (CAU)	-1,5,3	0	0	1	1	3.3%
Normal (NOR)	0,1	2	2	3	7	23%
Normal (NORF)	0,1	2	1	1	4	13.3%
Normal (NORP)	0,1	1	0	0	1	3.3%
Exponencial (DEXP)	1	1	1	1	3	10%
Beta (BETA)	5,3	0	0	0	0	0%
Gama (GAMA)	1,5,1	0	0	1	1	3.3%
Gama (GAMAD)	2,1	1	0	0	1	3.3%
Triangular (TRI)	0,1,3	2	1	0	3	10%
J1 -cuadrada (J1)	10	0	1	0	1	3.3%

n = Tamaño de muestra
Confiability 90%

**RESULTADOS DE LAS PRUEBAS DE BONDAD DE AJUSTE
PARA LOS GENERADORES DE MONTE CARLO
(Distribuciones Continuas)**

Distribución y generador	Parámetros	Rechazos n=100	Rechazos n=500	Rechazos n=1000	Total	Porcentaje
Uniforme (UNI)	0,1	0	1	0	3	3.3%
Cauchy (CAU)	-1,3,3	0	0	0	0	0%
Normal (NOR)	0,1	1	1	1	3	10%
Normal (NORF)	0,1	1	0	1	2	6.7%
Normal (NORP)	0,1	1	0	0	1	3.3%
Exponencial (DEXP)	1	1	1	1	3	10%
Beta (BETA)	5,3	0	0	0	0	0.0%
Gama (GAMA)	1,3,1	0	0	0	0	0.0%
Gama (GAMAD)	2,1	0	0	0	0	0.0%
Triangular (TRI)	0,1,3	1	0	0	3	10%
Ji-cuadrada (JI)	10	0	0	0	0	0.0%

n = Tamaño de muestra
Confiabilidad 95%

5.3.3 Conclusión de las pruebas.

De las tablas anteriores, podemos concluir que, exceptuando al generador NOR, no existe evidencia estadística para concluir que los generadores no son buenos. Esto, ya que el porcentaje de rechazo al 90% y 95% de confiabilidad, en estos generadores, no pasó del 10%.

En el caso del generador de NOR, el cual es el generador de normales descrito en el ejemplo de la sección 4.3.3 del capítulo IV, podemos concluir que no es un generador muy eficiente. Sin embargo, si el usuario no requiere de gran exactitud en su simulación, puede usar NOR ya que genera las muestras mucho más rápido que los otros generadores de normales.

5.4 Aleatoriedad.

En el caso de los generadores de números aleatorios de Monte Carlo, todas las distribuciones se generan a partir de una uniforme en $[0,1]$, en general por medio de funciones biyectivas. Por esto, sólo sería necesario realizar las pruebas de aleatoriedad para esta distribución. Sin embargo, como IMSL utiliza el método congruencial para genera las sucesiones de uniformes en $[0,1]$, el cual ha sido probado ampliamente por muchos matemáticos, entonces no se realizaron estas pruebas.

Las pruebas de aleatoriedad más utilizadas son: La prueba de los promedios, la prueba de frecuencias, la prueba de la distancia y la prueba de series.

Si el lector desea saber más sobre pruebas de aleatoriedad puede consultar los siguientes libros: W. J. Conover "Practical Nonparametric Statistics" [7], George W. Snedecar, William G. Cochran "Statistical Methods" [40], Bernard Ostle "Statistics in Research" [28].

CAPITULO VI Ejemplo Práctico.

6.1 Introducción

El objetivo de este capítulo es el de mostrar la simulación de un modelo de inventarios, demostrando así que el módulo de Monte Carlo tiene un uso práctico en diferentes campos de la planeación e investigación.

El ejemplo se tomó del libro Técnicas de Simulación en Administración y Economía de Robert C. Meier, William T. Newell y Harold L. Pazer, capítulo 1 y 9 [32].

En el libro, el modelo es primeramente determinístico y posteriormente los autores incluyen elementos estocásticos. Para efectos de este capítulo, sólo nos interesará la parte del modelo estocástico.

6.2 Modelo de Inventario.

Los autores nos explican que el ejemplo es un problema elemental de inventarios que "ilustra el modo en que puede simularse un fenómeno común que se encuentra en los sistemas de logística, distribución y administración de materiales".

Un artículo simple de inventario debe solicitarse bajo un sistema de punto de reorden y de cantidad de pedido. El artículo cuesta 5 dólares por unidad para la compra y se vende a 10 dólares cada uno. Los costos de mantenimiento del inventario, que son del 20% anual, se calculan de acuerdo con el valor (precio de compra) del inventario promedio. El costo de pedido y recepción de cada orden, para completar las existencias, es de 20 dólares independientemente del tamaño del pedido. El valor inicial del inventario es de 100 unidades. El punto de reorden por ahora, es de 40 unidades, es decir que cuando el inventario baje de este nivel se hará otro pedido. La cantidad del pedido es de 316 unidades. El modelo tiene 100 periodos de simulación.

El elemento estocástico del problema consiste en lo siguiente: Se supondrá que sólo se produce diariamente una demanda de artículo, aun cuando el número de unidades demandadas varía aleatoriamente, de acuerdo con la distribución de la siguiente tabla.

Tabla 2.1

Número de unidades demandadas	Probabilidad de ocurrencia
5	0.01
6	0.03
7	0.06
8	0.11
9	0.19
10	0.31
11	0.17
12	0.07
13	0.03

Demanda promedio = 9.72 unidades/día

El tiempo de espera de la recepción del pedido varía conforme a la distribución de la siguiente tabla

Tabla 2.2

Tiempo de espera	Probabilidad ocurrencia
2	0.15
3	0.20
4	0.30
5	0.20
6	0.15

Tiempo de espera promedio = 4 días

Se supondrá también que hay ciertas probabilidades, como se muestra en la siguiente tabla 2.3, de pérdida de órdenes que se retiran debido a la escasez de inventario.

Tabla 2.3

Número de períodos de retrasos	Probabilidad de pérdida
0	0.30
1	0.40
2	0.35
3	0.75
4	1.00

La tabla debe interpretarse en sentido que hay un 30% de probabilidades de perder una orden por cancelación en el día que se recibe, si no se dispone de inventario; 40% de probabilidades de perder un pedido que se retrasa al día siguiente, etc., hasta el quinto día, cuando hay un 100% de probabilidades de pérdida por un pedido no surtido en ese tiempo.

Cuando un pedido no puede surtir por falta de inventario y aún quedan artículos en el inventario, se surten todos los artículos que quedan y el remanente o bien se pierde o pasa a formar un pedido pendiente, que se surtirá cuando haya existencias.

Supóngase también que cuesta 10 dólares registrar un nuevo pedido cuando no puede surtir la demanda, y que la pérdida de órdenes, debida a la falta de existencias, es de 5 dólares por concepto de margen de beneficio unitario.

El objeto del modelo es de hallar el punto de reorden y la cantidad de pedido que reduzcan al mínimo el costo anual total de ese artículo.

Para efectos del ejemplo de este trabajo, se hará un análisis de sensibilidad tratando de encontrar un punto de reorden y una cantidad de pedido que minimice en lo posible el costo total de la operación del sistema.

6.3 Definición del Modelo en SAI.

6.3.1 Problemas de implementación del modelo.

Al tratar de modelar este ejemplo en el módulo de Monte Carlo se presentaron dos problemas. Uno fue que el modelo tiene periodicidad diaria y SAI no tiene forma de definir series diarias. El otro problema fue que no había forma de simular la parte del modelo correspondiente al cálculo de órdenes perdidas, el cual se describe en la tabla 2.3.

El primer problema se resolvió introduciendo una variable exógena correspondiente a los periodos de simulación y olvidando la periodicidad real del modelo. De hecho el modelo fue simulado en forma semestral, con inicio 102 y con fin 5101, lo que corresponde a 100 periodos de simulación.

Para resolver el segundo problema, fue necesario crear una función adicional llamada ORDEN que formará parte de la librería de funciones de Monte Carlo. Esta adición, fue relativamente sencilla y tuvo la ventaja de que a partir de ese momento, la función queda al alcance de todos los usuarios que tengan algún problema similar.

La función ORDEN tiene cinco argumentos x , l , vp , v y p . Si en el periodo t , x es menor o igual a l , entonces a partir de t y hasta $t+p$, ORDEN vale v con probabilidad $vp(n+1)$, donde vp es un vector de probabilidades y n es el número de periodos que faltan para llegar a $t+p$. En cualquier otro caso ORDEN vale cero. Como en la función ENTREGA si entre t y $t+p$, si x llega a ser mayor que l , el proceso es abortado.

En el caso de este ejemplo, la función ORDEN sirvió para definir la variable PEDAT (pedidos atrasados). Al definir esta variable, el tamaño del inventario corresponde a x , el punto de reorden a l , el valor de las órdenes no despachadas por falta de inventario a v , y el tiempo de espera del pedido como p . Como vector de probabilidades, se creó a PAT que contiene como elementos l menos la probabilidad de pérdida de un pedido de la tabla 2.3, ya que ésta es la probabilidad de que una orden no despachada, lo sea cuando el inventario la pueda cubrir.

6.3.2 Variables del modelo.

6.3.2.1 Variables endógenas. A continuación se presenta una tabla con los nombres de las variables endógenas que se utilizan en la simulación del modelo en SAI y una pequeña descripción de lo que representan.

Nombre de la variable	Significado
INV	Tamaño del inventario.
MINV	Inventario promedio.
COMPRA	Compra a proveedor.
VENTA	Venta a clientes.
TIES	Tiempo de espera para un pedido.
DEM	Demanda diaria.
DEMP	Demanda diaria perdida.
DEMTOT	Demanda diaria más órdenes atrasadas.
OND	Órdenes no despachadas.
PEDAT	Pedidos atrasados.
SPAT	Suma de pedidos atrasados.
COSMAN	Costos de mantenimiento.
COSCOM	Costos de compra.
COSPER	Costo por ordenes perdidas.
COSPA	Costo de pedidos atrasados.
COSSUM	Acumulación de la suma de COSCOM, COSPER Y COSPA.
COSTOT	Costo total.

6.3.2.2 Variables exógenas. La única variable exógena del modelo es PER que contiene una rampa de 0 a 100 que corresponde a los períodos de simulación. Es decir, PER es una serie cuyos elementos son: 0, 1, 2, 3,, 99, 100.

6.3.2.3 Parámetros.

Nombre de los parámetros	Significado	Valor
REORDEN	Punto de reorden.	40
PEDIDO	Cantidad de pedido.	316
VTIE	Vector de valores de tiempo de espera.	Ver tabla 2.2
PTIE	Vector de probabilidades de ocurrencia de tiempo de espera.	Ver tabla 2.2
NT	Dimensión de VTIE y PTIE.	5
VDEM	Vector de valor de la demanda.	Ver tabla 2.1
PDEM	Vector de probabilidades de ocurrencia de los valores de la demanda.	Ver tabla 2.1
ND	Dimensión de VDEM y PDEM.	10
PAT	Vector de probabilidades de pérdida de un pedido no cubierto.	(0.7,0.6,0.45,0.25,0) (ver tabla 2.3)
TI	Tasa de interés.	0.08% diario o 20% anual (250 días).

6.3.2.4 Ecuaciones en SAI para el modelo. Para simular el modelo de inventario de este ejemplo, se tuvieron que definir 17 ecuaciones en el módulo de Monte Carlo. A continuación se presentan estas ecuaciones, cada una con una explicación de lo que representa.

1.-INV=INV(-1)+COMPRA-VENTA

El tamaño del inventario en t es igual al tamaño del inventario en t-1, más lo que se compra y menos lo que se vende.

2.-COMPRA=ENTREGA(INV(-1),REORDEN,PEDIDO,TIES)

Si en el período t-1, el inventario fue menor o igual a REORDEN, entonces se hace un pedido en t de cantidad de pedido de PEDIDO, que es entregado en el período t+TIES.

3.-VENTA=DEMTOT+MIN(INV(-1)-DEMTOT,0)

La venta en un período t es igual a la demanda total si alcanza el inventario. En caso contrario, la venta es igual al valor del inventario. La venta está formada por la demanda diaria y por los pedidos atrasados que no han sido despachados.

4.-TIES=ARBOL(VTIE,PTIE,NT)

El tiempo de espera de recepción de pedido es una variable aleatoria que sigue la distribución definida en la tabla 2.2

5.-DEMTOT=DEM+SPAT(-1)

La demanda total es igual a la demanda diaria más la suma de pedidos atrasados.

6.-DEM=ARBOL(VDEM,PDEM,ND)

La demanda diaria es una variable aleatoria cuya distribución está dada en la tabla 2.1

7.-SPAT=(SPAT(-1)+PEDAT)*MAY(OND,0)

La suma de pedidos atrasados en t es igual a la suma de pedidos atrasados en t-1 más los pedidos atrasados, siempre y cuando la variable OND sea positiva. Lo último quiere decir simplemente que cuando los pedidos atrasados ya han sido despachados, entonces la suma de pedidos atrasados debe ser cero.

8.-PEDAT=ORDEN(INV(-1),REORDEN,PAT,OND,TIES)

Las órdenes no despachadas pueden pasar a ser pedidos pendientes con la probabilidad que se describe en la tabla 2.3

9.-DEMP=OND-PEDAT

La demanda perdida es igual a las órdenes no despachadas menos los pedidos que se quedan pendientes.

10.-OND=(-MIN(INV(-1)-DEM,0))

La variable "órdenes no despachadas" es igual a cero si el inventario alcanza a cubrir la demanda total. En caso contrario, la variable es igual a la diferencia de la demanda y del inventario.

$$11.-COSTOT=COSSUM+COSMAN$$

El costo total acumulado es igual a la acumulación de los costos diarios más el costo de mantenimiento.

$$12.-COSCOM=VENTA*5+20*MAY(COMPRA,0)$$

El costo de compra es igual a la venta por cinco que es el valor de compra de un artículo, más 20 en el caso de que se haya hecho algún pedido al proveedor.

$$13.-COSPER=5*DEMP$$

Costo por órdenes perdidas es igual a la demanda perdida por 5 que es el concepto de margen de beneficio unitario.

$$14.-COSMAN=MINV*5*TI*PER$$

Costo de mantenimiento es igual a la tasa de interés por período (0.0008) por el número de períodos, por el valor de compra del inventario promedio.

$$15.-COSPA=10*MAY(PEDAT,0)$$

El costo por pedidos atrasados es igual a 10 cuando hubo pedidos atrasados

$$16.-MINV=(MINV(-1)*PER(-1)+INV)/PER$$

El inventario promedio es calculado siguiendo la siguiente fórmula

$$\bar{x}_i = \frac{n-1}{n} \sum_{j=1}^{i-1} x_j + \frac{x_i}{n}$$

$$17.-COBSUM=COBSUM(-1)+COSPA+COSCOM+COSPER$$

COBSUM es la acumulación de los costos diarios a través de la simulación.

6.3.3 Ejemplo de una corrida de simulación.

A continuación se presenta una tabla con los resultados de una de las corridas de simulación. En ella el lector puede ver el comportamiento que tuvo el inventario, las órdenes de restitución, la venta, la demanda real, los pedidos pendientes, la demanda perdida y el costo total de la operación durante los cien períodos de la corrida.

Tabla 3.1

RESULTADOS DE LA SIMULACION (Ejemplo de una corrida)

período	Exis- tencia	Compra	Venta	Desan- da	Atra- sados	Perdi- dos	Costo total
0	100						
1	89	0	11	11	0	0	55
2	79	0	10	10	0	0	106
3	66	0	13	13	0	0	171
4	53	0	13	13	0	0	236
5	43	0	10	10	0	0	286
6	33	0	10	10	0	0	336
7	27	0	6	6	0	0	367
8	18	0	9	9	0	0	412
9	8	0	10	10	0	0	467
10	0	0	8	10	0	2	512
11	0	0	0	12	0	12	572
12	0	0	0	10	0	10	622
13	316	316	0	10	0	10	693
14	307	0	9	9	0	0	739
15	298	0	9	9	0	0	785
16	280	0	10	10	0	0	837
17	279	0	9	9	0	0	883
18	266	0	12	13	0	0	949
19	255	0	11	11	0	0	1005
20	242	0	13	13	0	0	1071
21	233	0	9	9	0	0	1117
22	223	0	10	10	0	0	1167
23	212	0	11	11	0	0	1223
24	202	0	10	10	0	0	1274
25	189	0	13	13	0	0	1340
26	184	0	5	5	0	0	1366
27	176	0	8	8	0	0	1404
28	166	0	10	10	0	0	1457
29	157	0	9	9	0	0	1502
30	147	0	10	10	0	0	1553
31	137	0	10	10	0	0	1604
32	126	0	11	11	0	0	1659
33	113	0	13	13	0	0	1725
34	103	0	10	10	0	0	1775
35	90	0	12	12	0	0	1841
36	77	0	13	13	0	0	1906
37	67	0	10	10	0	0	1956
38	57	0	10	10	0	0	2006
39	44	0	13	13	0	0	2071

40	34	0	10	10	0	0	2122
41	27	0	7	7	0	0	2157
42	14	0	11	11	0	0	2212
43	5	0	11	11	0	0	2267
44	316	316	5	9	0	4	2333
45	306	0	10	10	0	0	2384
46	293	0	13	13	0	0	2450
47	285	0	8	8	0	0	2492
48	273	0	12	12	0	0	2553
49	261	0	12	12	0	0	2614
50	251	0	10	10	0	0	2665
51	244	0	7	7	0	0	2701
52	235	0	9	9	0	0	2747
53	222	0	13	13	0	0	2813
54	215	0	7	7	0	0	2848
55	206	0	9	9	0	0	2894
56	198	0	8	8	0	0	2935
57	188	0	10	10	0	0	2986
58	175	0	13	13	0	0	3051
59	162	0	13	13	0	0	3117
60	149	0	13	13	0	0	3183
61	139	0	10	10	0	0	3233
62	129	0	10	10	0	0	3284
63	116	0	13	13	0	0	3342
64	103	0	13	13	0	0	3415
65	93	0	10	10	0	0	3485
66	81	0	12	12	0	0	3525
67	71	0	10	10	0	0	3576
68	62	0	9	9	0	0	3621
69	49	0	13	13	0	0	3686
70	40	0	9	9	0	0	3731
71	33	0	7	7	0	0	3766
72	23	0	10	10	0	0	3816
73	14	0	9	9	0	0	3862
74	3	0	11	11	0	0	3917
75	316	316	3	13	10	0	3963
76	294	0	20	10	0	0	4044
77	286	0	10	10	0	0	4115
78	277	0	9	9	0	0	4161
79	267	0	10	10	0	0	4212
80	258	0	9	9	0	0	4258
81	248	0	10	10	0	0	4309
82	235	0	13	13	0	0	4375
83	225	0	10	10	0	0	4426
84	215	0	10	10	0	0	4477
85	206	0	9	9	0	0	4523
86	197	0	9	9	0	0	4569

87	188	0	9	9	0	0	4615
88	175	0	13	13	0	0	4680
89	166	0	9	9	0	0	4726
90	159	0	7	7	0	0	4761
91	147	0	12	12	0	0	4822
92	134	0	13	13	0	0	4888
93	121	0	13	13	0	0	4953
94	108	0	13	13	0	0	5018
95	99	0	9	9	0	0	5064
96	87	0	12	12	0	0	5124
97	74	0	13	13	0	0	5189
98	61	0	13	13	0	0	5255
99	48	0	13	13	0	0	5320
100	36	0	12	12	0	0	5380

Costo total de esta corrida = 3380. Demanda total perdida = 36

6.3.4 Análisis de sensibilidad.

Este análisis tiene por finalidad medir qué tan sensible es el costo total de operación del artículo, al modificar el punto de reorden y la cantidad de pedido, y tomar los puntos en los cuales el costo total fue mínimo.

El número de corridas fue igual a cien, puesto que al realizar la simulación con 500 corridas, se encontró que las estimaciones de medias y desviaciones estándar, eran prácticamente las mismas.

A continuación, se presenta una tabla que describe el análisis de sensibilidad que se realizó a este modelo. Las dos primeras columnas corresponden a los parámetros que cambian. Las otras describen la estimación del costo total. Esta estimación consiste en la media, la desviación estándar y de un intervalo de confianza. Este último se calcula restándole y sumándole a la media el doble de la desviación estándar. Si suponemos que el costo tiene una distribución normal, este intervalo de confianza tiene confiabilidad de aproximadamente 90%.

Tabla 3.2

ANALISIS DE SENSIBILIDAD

Punto de reorden	Cantidad de pedido	Media del costo	Desviación estándar	Límite Inferior	Límite Superior
40	316	5331.87	101.62	5128.58	5535.11
5	100	5380.60	110.05	5160.50	5600.70
10	200	5325.40	109.00	5107.40	5543.40
30	300	5331.64	100.32	5131.00	5532.28
50	400	5342.39	100.52	5141.35	5543.43
80	316	5340.80	101.00	5138.00	5542.00
300	600	5194.00	103.60	4986.80	5401.20
400	700	5492.26	97.72	5296.82	5688.30
350	650	5217.90	105.09	5007.72	5428.08
250	550	5205.70	110.28	4985.14	5426.26
500	800	5552.75	97.95	5356.85	5748.65

6.3.5 Conclusiones.

La desviación estándar permanece más o menos constante a través del análisis, por lo que para determinar qué punto fué el que produjo menos costo, no es necesario hacer pruebas de hipótesis formales.

El punto donde se obtuvo el costo menor en este pequeño análisis, es el de punto de reorden de 300 y cantidad de pedido 600. Muy probablemente, el punto óptimo del sistema se encuentra cerca de este punto. Habría que tomar en cuenta más puntos para acercarse al óptimo.

Como puede observar el lector, la curva que sigue el costo total de operación del artículo con respecto al punto de reorden y cantidad de pedido, es bastante plana. Realmente, no hubo cambios muy significativos al modificar los parámetros. Sin embargo, si este modelo fuera, por ejemplo, un modelo de inventario de granos de un país, y las cifras estuvieran dadas en miles de dólares, entonces este análisis podría ahorrarle al gobierno alrededor de 350,000 dólares, lo cual es significativo.

CAPITULO VII Conclusiones

Un sistema con las características de SAI ofrece al usuario la posibilidad de escoger dentro de una gran cantidad de alternativas la posible solución a su problema. El sistema es flexible y es una herramienta poderosa de fácil uso. Es por esto que es un sistema conveniente para la investigación económica, financiera y estadística.

Además, el sistema ha sido muy útil para estudiantes de diferentes carreras, por lo que su utilización en áreas académicas sería de gran beneficio.

La simulación en computadora es cada día más utilizada como una herramienta para la investigación y la planeación en diferentes campos. Es por esto que un sistema como SAI necesita contar con la posibilidad de simular modelos.

Monte Carlo es una de las técnicas de simulación que ha tenido un gran desarrollo en los últimos años. Con este método es posible solucionar tanto problemas estocásticos como determinísticos, estimando un parámetro de una población conocida o supuesta.

En SAI, el módulo de simulación Monte Carlo, simula modelos de series de tiempo con esta técnica. Sirviéndose de las funciones aleatorias que están definidas en el módulo, el usuario introduce el elemento aleatorio de su modelo.

El usuario de SAI, puede hacer evolucionar un modelo de Monte Carlo fácilmente, el sistema le proporciona herramientas para construir distintos escenarios, para modificar el modelo o el tiempo de simulación y para interpretar los resultados rápidamente.

Dado que el sistema se apoya grandemente en la librería IMSL, la cual posee gran reconocimiento entre matemáticos y estadísticos, el usuario de SAI puede estar seguro que obtendrá buenos resultados en sus simulaciones.

El sistema es robusto porque provee un gran número de posibilidades en el diseño de los modelos, permitiendo así, la solución de un gran cantidad de problemas.

La programación del sistema está hecha de tal forma que es relativamente sencillo hacerle modificaciones. Esto hace que el sistema pueda mejorarse sensiblemente, tomando en cuenta la retroalimentación de los usuarios. Por esto, el sistema Monte Carlo se encuentra siempre en desarrollo.

El sistema es muy versátil, puesto que permite modificar la definición de un modelo en el tiempo. Además, el manejo de expresiones aritméticas, la librería de funciones predefinidas y la posibilidad del usuario de definir funciones propias al modelo, hace de Monte Carlo un sistema poderoso.

NOTA FINAL DEL COMITÉ DE LA COMISIÓN

SAI, y especialmente el módulo de Monte Carlo, es una gran ayuda para aquellos tomadores de decisiones que simulan con la técnica de Monte Carlo. Les permite olvidarse del trabajo computacional que implica generalmente este proceso. Dedicando, así, este tiempo a lograr un análisis más profundo del problema, traduciéndose en una mejor toma de decisiones.

Convertir lo complicado en sencillo es una cualidad del hombre moderno, que apoyándose en adelantos como SAI y su módulo de simulación Monte Carlo, puede tomar decisiones ahora, para un mejor futuro.

APENDICES

B.1 Apéndice A

Sintaxis del módulo de simulación Monte Carlo del sistema de análisis interactivo SAI.

Para describir la sintaxis de comandos y atributos del sistema, se utilizan las siguientes convenciones:

- Las palabras en mayúscula indican palabras que deben ir textualmente en el comando o atributo. - Las palabras en minúsculas entre llaves (()) indican valores que tienen que ser remplazados por elementos como son nombres de variables, archivos, números, etc.

- Los elementos que se encuentren entre corchetes ([]), son elementos opcionales, que irán o no dependiendo de lo que el usuario desee que el sistema haga.

- Los puntos suspensivos, indican que el elemento anterior se puede repetir un número indeterminado de veces.

- Cuando un elemento tiene varias alternativas, estas van una debajo de la otra.

B.1.1 Comandos y atributos generales del sistema.

En esta sección veremos los comandos y atributos que el usuario puede utilizar en cualquier momento, independientemente de la fase en que se encuentre.

Existen comandos en Monte Carlo que son generales para SAI. Estos son el comando FIN, el comando SALIDA y el comando AYUDA. FIN sirve para dejar el módulo y regresar al módulo central de SAI, SALIDA sirve para asignar la salida de listados y diagnósticos a la pantalla o a la impresora, y AYUDA despliega en la pantalla ayudas sobre el manejo del sistema y de sus comandos y atributos. Estos comandos tienen la misma sintaxis que en todo SAI. Por esto no se incluye en este apéndice. El lector puede consultar el manual del usuario de SAI para mayor información al respecto.

El signo de "\$" representa la orden de ejecutar un comando en cualquiera de las fases del sistema. Una vez definido un modelo en la Fase Completa, el usuario debe dar este signo, para que el proceso de simulación de comienzo. Lo mismo pasa en la Fase Datos, Fase Resultados y Fase diagnóstico.

B.1.1.1 Comando NOMBRE.- Este comando sirve para definir el nombre de un modelo o para referirse a uno ya existente. Sintaxis: NOMBRE (nombre modelo) [; NUEVO] \$

donde (nombre modelo) es el nombre del modelo.

El nombre puede tener hasta ocho caracteres, siendo obligatorio que empiece con una letra.

Al definir un modelo por primera ocasión, el usuario debe de utilizar la opción "; NUE". Posteriormente, para referirse al modelo una vez ya definido, el analista debe omitir esta opción, de otra manera, el sistema pierde el modelo.

Ejemplo.-

1) NOM PETROL;NUE*

(Como en todo SAI, es suficiente poner las tres primeras letras de comandos y argumentos.)

En este ejemplo, el usuario determina que el nombre de su modelo es PETROL y además, que PETROL es un modelo nuevo que apenas se va a definir.

2) NOM PETROL*

En este caso, el usuario va a trabajar con el modelo PETROL que definió en alguna sesión anterior.

B.1.1.2 Atributo FASE.- Este atributo sirve para que el usuario especifique al sistema que fase del módulo desea utilizar. El usuario tiene sólo cuatro opciones que son: Completa, Datos, Resultados y Diagnóstico.

En general el usuario no necesita especificar la fase, puesto que el sistema se da cuenta de acuerdo al contexto de otras instrucciones. No obstante, en ciertos casos, el usuario debe de especificarla.

Sintaxis: FASE [COMPLETA] ; DATOS RESULTADOS DIAGNOSTICO

Ejemplos.-

1) FASE COMPLETA;

El usuario va a definir un modelo nuevo, o bien, a redefinir uno ya existente.

2) FASE RES;

El analista quiere que el sistema recupere los resultados de la última simulación del modelo.

B.1.1.3 Atributo TIEMPO.- Este comando define el intervalo de tiempo en que el sistema efectuará la simulación del modelo.

Sintaxis: TIEMPO (periodicidad),(fecha de inicio),(fecha final); donde (periodicidad) es un número entero que corresponde al número de periodos que caben en un año, (fecha de inicio) es la fecha en que comienza la simulación y (fecha final) en la que finaliza. Tanto la fecha inicial como la final son números enteros de cuatro dígitos, en los cuales las dos primeras cifras corresponden al año y las dos últimas al periodo dentro del año.

Ejemplos.-

1) TIE 1,7801,8501;

Aquí, el modelo es anual, empieza en 1978 y acaba en 1985. (Cuando un modelo es anual, las últimas dos cifras de las fechas deben ser siempre 01).

2) TIE 4,7702,8604;

En este caso, el modelo es trimestral (hay cuatro trimestres en un año), empieza en el segundo trimestre de 1977 y termina en el último trimestre de 1986.

B.1.1.4 Atributo CORRIDAS.- Con este atributo, el usuario define el número de ocasiones en que se evaluará el modelo en la simulación.

Sintaxis: CORRIDAS (número de corridas) ;

donde (número de corridas) es un cualquier número entero positivo menor o igual que 32767.

Ejemplo.-

CORR 1500;

El modelo se evaluará 1500 veces.

B.1.1.5 Comando DESPLIEGA.- Con este comando, el usuario obtiene un listado de ecuaciones o funciones del modelo.

Sintaxis: DESPLIEGA [ECUACION] [(ini) [(fin)]] * FUNCION

donde (ini) es el número de la ecuación con la que el usuario quiere empezar el listado y (fin) en número de la última.

La opción "ECUACION" es para listar ecuaciones y la opción "FUNCION" para funciones.

En caso de que el usuario omita (ini) y (fin), el sistema listará todas las ecuaciones (funciones) del modelo. Si la omisión es sólo de (fin), el sistema listará únicamente la ecuación (función) correspondiente a (ini).

Ejemplos.-

.

1) DES ECU;

Esta instrucción hace que el sistema liste todas las ecuaciones.

2) DES FUN 3,8*

El sistema listará de la ecuación 3 a la 8 inclusive.

3) DES ECU 5*

quiere decir, despliega la ecuación 5.

B.1.1.6 Comando BORRA.- Con este comando el usuario puede borrar un conjunto de ecuaciones o funciones de su modelo.

Sintaxis: BORRA [ECUACION] [(ini) [, (fin)]] * FUNCION

Como se puede ver, la sintaxis de BORRA es equivalente a la de despliga, por lo que no es necesario explicarla de nuevo.

Ejemplo.-

1) BORRA FUN 3,7*

Al dar esta instrucción, el sistema borra de la función 3 a la 7 inclusive.

B.1.2 Comandos y atributos específicos de la Fase Completa.

B.1.2.1 Atributo ECUACION.- Para definir una ecuación, el analista utiliza este atributo. Cada ecuación tiene asociado un número para distinguirla. El usuario puede dar la ecuación en tres rengiones y ésta puede estar formada hasta por 200 caracteres.

Sintaxis:

ECUACION (número ecu), '(reng 1)'
, '(reng 2)'
, '(reng 3)'';

Donde (número ecu) es un número entero positivo correspondiente al número de la ecuación, (ecuación) es la ecuación que define el usuario (ver capítulo III) y (reng 2) y (reng 3) son respectivamente el segundo tercer rengiones de continuación de la ecuación.

Ejemplos.-

1) ECU 1, 'UTI=VENTAS-COSTOS-GASTOS';

El usuario define la ecuación 1 de su modelo como

UTI=VENTAS-COSTOS-GASTOS.

2) ECU 4, 'TRG=KLM^4+COS(LGN(PBR))',
'-YTF=0.5+PIS=TRG(-1)+',
'EXP(PBR/2)';

En este caso la ecuación 4 es:

$$\text{TRG}=\text{KLM}^4+\text{COS}(\text{LGN}(\text{PBR}))-\text{YTF}^*0.5+\text{PIS}=\text{TRG}(-1)+\text{EXP}(\text{PBR}/2)$$

8.1.2.2 Atributo FUNCION.- Para definir las funciones, el analista se vale del atributo FUNCION. Al igual que las ecuaciones, cada función tiene asociada un número para distinguirla y puede estar dada hasta en tres renglones. Igualmente, puede tener hasta 200 caracteres.

Sintaxis:

FUNCION (número fun), '(función)' [, '(reng 2)'] [, '(reng 3)']
]; donde (número fun) es un número positivo que distingue a la función, (función) es la definición de la función (ver capítulo III) y (reng 2) y (reng 3) son respectivamente el segundo y el tercer renglón de continuación de la ecuación.

Ejemplos.-

1) FUN 1, 'NORMA(X,Y)=X^2+Y^2';

Esto define a la función 1 del modelo, llamada "NORMA", para que calcule la norma de un vector de dos componentes (x,y).

2) FUN 2, 'PTK(A,B,C,D,E),(7701;8301)=5*A/TYB+B*TYB(-1)',
'+C*LGN(GTX)';

al dar esta instrucción el usuario define la función

PTK(A,B,C,D,E),(7701;8301)=5*A/TYB+B*TYB(-1)+C*LGN(GTX)

(El usuario al definir la función de esta manera, sólo la hace válida para el intervalo de tiempo entre 1977 y 1983.)

8.1.2.3 Atributo ESTADISTICAS.- Con este atributo, el usuario especifica sobre qué variables quiere que el sistema calcule estadísticas.

Sintaxis:

ESTADISTICAS

{(var),(med),(des)}[(var2),(med2),des2][..];

TODAS

NO

donde (var), (var2), ... son los nombre de las variables a las que el sistema calculará los estadísticos, (med), (med2),... son los nombres de las series donde el sistema guardará las medias y (des), (des2),... son los nombres de las series donde el sistema guardará las desviaciones estándar.

Si el usuario toma la opción "TODAS", el sistema calculará estadísticas para todas las variables del modelo. En este caso, los nombres de las series de estadísticas son formadas por el sistema, anteponiendo al nombre de la variable una "M" para las medias y una "D" para las desviaciones estándar.

Si el usuario desea cancelar el cálculo de estadísticas antes de la simulación, entonces debe utilizar la opción "NO".

Ejemplos.-

1) EST
PIB,MEDPIB,DESPIB,TAS,MEDTAS,DESTAS,INF,MEDINF,DESINF;

Al dar esto, el usuario indica al sistema que calcule estadísticas para las variables PIB, TAS y INF de su modelo. Los nombres de las series de medias son respectivamente MEDPIB, MEDTAS y MEDINF. Para las series de desviaciones estándar, los nombres son respectivamente DESPIB, DESTAS y DESINF.

2) EST TODAS;

En este caso, el sistema calculará estadísticas para todas las variables del modelo. Por ejemplo, si existe una variable llamada PIB en el modelo, entonces el sistema creará dos series, una llamada MPIB que contendrá las medias y la otra llamada DPIB que tendrá las desviaciones estándar.

8.1.3 Comandos y atributos específicos de la Fase Datos.

8.1.3.1 Atributo MODIFICA.- MODIFICA sirve para que el usuario especifique que variables modificó en el escenario del modelo, para efectuar una simulación en la Fase Datos. Si el analista no especifica ninguna variable, el sistema supondrá que modificó todas las variables del escenario, aunque éste no sea el caso.

Este atributo, en modelos grandes, ahorra tiempo a la ejecución de la Fase Datos. Sin embargo, el usuario puede omitir su uso, sin afectar en nada el proceso.

Sintaxis: MODIFICA (var1) [,(var2) [...]];

donde (var1), (var2),... son los nombres de las variables que modificó el usuario.

La opción "TODAS" nulifica alguna orden anterior de MODIFICA que haya dado el usuario. Es como decirle al sistema que todas las variables del escenario fueron modificadas.

Ejemplo.-

1) MOD PIB,INF,TASA,PTC;

Con esta instrucción, el usuario dice al sistema que modificó las variables PIB, INF, TASA y PTC del escenario de su modelo.

8.1.4 Comandos y atributos específicos de la Fase Resultados

8.1.4.1 Atributo MANTEN.- Con este comando, el usuario especifica que resultados quiere recuperar después de una simulación.

Si el sistema calculó estadísticas de alguna de las variables, que desea recuperar el usuario, el sistema automáticamente recuperará las variables de estadísticas.

Cuando el usuario no especifica ninguna variable para mantener, el sistema supone que el usuario desea recuperar todas las variables del modelo.

Sintaxis: MANTEN [(var1) [,(var2) [...]]]; TODAS

donde (var1), (var2),... son los nombres de las variables que el sistema recuperará.

La opción "TODAS" cancela alguna orden anterior de MANTEN. Es decir que recuperará todas las variables.

Ejemplo.-

MAN PRT,DEV,TOT;

El sistema recuperará las variables con nombres PRT, DEV y TOT con sus respectivas variables de estadísticas.

8.2 Apéndice B.- Funciones generadoras de números aleatorios.

8.2.1 Distribución Gama.

Función: GAMA.

Modo de empleo: GAMA(a,b).

Donde a es el parámetro de forma y b el parámetro de escala.

Algoritmo: Se generan muestras aleatorias con la densidad gama utilizando varios algoritmos computacionales, dependiendo del valor del parámetro a.

Si $1 > a$ y $a > 0.5$, es el utilizado un método de dos pruebas de rechazo debido a Ahrens.

Si $a > 0.5$, se toma la mitad de la raíz cuadrada de muestras aleatorias normales.

Si $a = 1$, la función toma muestras exponenciales.

Si $a > 1$, se usa un procedimiento de 10 regiones de rechazo que se describe en la referencia 2.

Referencia 1.- Robinson ,D.W.,and Lewis, P.A.W.,Generating Gama and Cauchy random deviates [32]. Referencia 2.-Schmeiser,Bruce W. and Lal R., "Squeeze methods for generating Gamma variates" [37].

B.2.2 Distribución Beta

Función: BETA.

Modo de empleo: BETA(p,q).

Donde p,q son los parámetros de la distribución beta.

Algoritmo: BETA genera muestras aleatorias con función de densidad beta, mediante el uso de cuatro diferentes algoritmos. Estos algoritmos son utilizados dependiendo de los parámetros p y q, y se describen en las referencias.

Referencias:

1.- Atkinson , A.,C., "A family of switching algorithms for the computer generation of beta random variates" [2].

2.-Cheng, R.C.H., "Generating beta variates with nonintegral shape parameters" [6].

3.-Johns,M.D., "Erzeugung von Betaverteilten und Gammaverteilten Zufallszahlen" [19].

4.-Schmeiser,Bruce W., and Babu,A.J.G., "Beta variate generation via exponential majorizing functions"[38].

B.2.3 Distribución Binomial.

Función: BIN

Modo de empleo: BIN(n,p)

Donde n es el número de experimentos y p es la probabilidad de éxito en un experimento.

Algoritmo: BIN genera una muestra binomial con parámetro n y p. Si n es mayor que 34, se utiliza un método de medianas de muestras aleatorias de una distribución uniforme. Las medianas se obtienen a partir de distribuciones aleatorias Beta. Este método se describe en la referencia. Si n es menor que 35, se usa un simple método de conteo.

Referencia.- Relles,Daniel A., "A simple algoritme for generating binomial random variables when n is large"[30].

B.2.4 Distribución Cauchy.

Función CAU.

Modo de empleo: CAU(a,b)

Donde a son los grados de libertad del numerador y b del denominador.

Algoritmo: La densidad de Cauchy tiene distribución

$$F(x) = 0.5 + \left(\frac{1}{\pi}\right) \tan^{-1}(x)$$

x real. por inversión, una muestra se representa como
 $x = \tan \pi(u - 0.5)$

donde u es una muestra uniforme en [0,1].

Referencia .- Robinson ,D.,W.,and Lewis, P.A.W.,Generating
Gamma and Cauchy random deviates [33].

8.2.5 Distribución Ji-Cuadrada.

Función JI

Modo de empleo: JI(n)

Donde n es el número de grados de libertad.

Algoritmo: JI genera muestras aleatorias con distribución
ji- cuadrada, utilizando una relación entre variables ji-
cuadradas y el producto de uniformes en [0,1]
independientes. Si u_1, u_2, \dots, u_m son variables aleatorias en [0,1]
independientes entonces

$$Y = -2 \left(\ln \left(\prod_{i=1}^m u_i \right) \right)$$

tiene una distribución ji-cuadrada con con 2m grados de
libertad. El valor de m es $n/2$ o $(n-1)/2$ si n es par o impar
respectivamente.

8.2.6 Distribución Geométrica.

Función: GEOM

Modo de empleo: GEOM(p)

Algoritmo :Las muestras con distribución Geométrica se
generan igualando x con la primera n tal que $r(n)$ (muestras
uniforme en [0,1]) sea menor o igual a p.

8.2.7 Distribución Exponencial.

Función: DEXP

Modo de empleo: DEXP(l)

Donde l es la media de la distribución.

Algoritmo:Las muestras Exponenciales, se generan mediante la
función inversa de la función de distribución exponencial.
Así, $x = -l(\ln(u))$, donde u es una muestra aleatoria uniforme en
[0,1] tiene distribución exponencial.

Nota.- Esta función no utiliza la librería IMSL.

8.2.8 Distribución Poisson.

a) Función: POIS

Modo de empleo: POIS(l)

Donde l es el parámetro de la distribución poisson.

Algoritmo: POIS genera muestras aleatorias poisson. La resultante k, es el número más chico m tal que r es menor o igual a

$$\sum_{j=0}^m (e^{-l}) \frac{l^j}{j!}$$

donde r es uniforme [0,1].

Nota: Esta función se usa en el caso de que el parámetro l no cambia muy seguido.

Referencia.- Snow, Richard H., "Algorithm 342, generator of random satisfying the poisson distribution [41].

b) Función :POISD

Modo de empleo: POISD(l)

Algoritmo: La resultante k es tal que r_1, \dots, r_n sea mayor que $\exp(-l)$. r_1, r_2, \dots, r_n son uniformes en [0,1]. Si l es mayor que 50 se utiliza una aproximación normal.

Nota.- Esta función se utiliza en caso de que l varíe mucho.

Referencia: Schaffer, Henry E., "Algorithm 369, generator of random numbers satisfying the Poisson distribution" [35].

8.2.9 Distribución Normal

a) Función: NOR

Modo de empleo: NOR(m,v)

Donde m es la media y v la varianza de la distribución.

Algoritmo: Para generar muestras normales con media cero y varianza 1, se toma la suma de doce muestras uniformes en [0,1] menos 6. Por el teorema central del límite, esta suma se aproxima a una NOR(0,1). Posteriormente se multiplica por la desviación estándar y se le suma la media para obtener la muestra NOR(m,v). Se toman doce muestras ya que se obtiene una buena aproximación, con pocos cálculos.

Nota.- Esta función no utiliza librería IMSL.

Referencia: Diego Bricio Hernández y Luis Javier Alvarez. Informe Monográfico, "El Método Monte Carlo" [15].

b) Función: NORFI

Modo de empleo: NORFI(m,v)

Algoritmo: NORFI genera las muestras, usando el método de inversión de la función de distribución normal. Esta función inversa es aproximada.

c) Función: NORP

Modo de empleo: NORP(m,v)

Algoritmo: NORP genera muestras aleatorias normales por el método polar, descrito en la referencia. Este algoritmo se debe a G.E.P. Box, M.E. Muller, and G. Marsaglia. Dos muestras independientes uniformes en [-1,1] son generadas, hasta que la suma de sus cuadrados (s) sea menor que 1. Dos muestras normales se forman como el producto de cada uniforme y la raíz cuadrada de

$$(-2\text{Log}(s))/s$$

Referencia : Knuth, Donald E., "The art of Computer Programming" [20].

B.2.10 Distribución Triangular.

Función: TRI

Modo de empleo: TRI(a,b,c)

Donde a es el valor mínimo, b el valor más probable y c el valor máximo.

Algoritmo: Las muestras se generan por inversión de la función de distribución triangular y valuándola en muestras aleatorias uniformes en [0,1].

Nota: Esta función no utiliza la librería IMSL.

B.2.11 Distribución Uniforme.

a) Función UNI

Modo de empleo: UNI(a,b)

Donde [a,b] es el intervalo muestral.

Algoritmo: Dada una semilla s, las muestras $r(i)$ $i=1,2,\dots,n$ se generan por:

$$s(0) = s$$

$$s(i) = 7^i s(i-1) \pmod{2^{31} - 1}$$

$$r(i) = 2^{-31} s(i)$$

Referencias:

1.- Lewis, P.A.W., GOODMAN, A.S., AND MILLER, J.M., "PSEUDO-RANDOM GENERATOR FOR THE SYSTEM/360" [21].

2.- Learmonth, G., P. and Lewis, P.A.W., "Naval Postgraduate School Random Number Generator Package LLRANDOM" [22].

3.-Learmonth, G.P., and Lewis, P.A.W., "Statistical tests of some widely used and recently proposed uniform random number generators " [23].

b) Función: UNID

Modo de empleo: UNID(a,b)

Nota .- Esta función genera números enteros uniformemente distribuidos en el intervalo [a,b].

Algoritmo: Se toma una muestra u uniforme en [0,1], y se calcula $x = [ru + (a-b)] + a$, donde [r] es la parte entera de r.

B.3 Apéndice C.- Ejemplo de una sesión de un usuario en el módulo de Monte Carlo.

Para ilustrar una sesión de trabajo en SAI, donde el usuario define un modelo, se tomó como ejemplo el modelo de inventarios del capítulo VI.

Para este ejemplo, supondremos inicialmente que el usuario se encuentra en el módulo de control de SAI y supondremos también que ya generó el banco de datos del escenario y que lo definió como área de trabajo.

A continuación se presenta esta sesión, enseñando la forma como la vería un usuario en caso de que definiera en realidad el modelo. En un renglón, todo lo que sigue después del signo ">", corresponde a las instrucciones que da el usuario. Las frases que se encuentran entre signos de admiración, son comentarios explicativos del ejemplo.

B.3.1 Sesión de la definición del modelo de inventarios.

```
FUNCION >EJE MONS ! El control pasa al módulo Monte Carlo !
```

```
S.A.I. MODULO DE EJECUCION DE MONTECARLO VERSION 3.1
```

```
MONTECARLO >NOM INV;NUE ! El nombre del modelo es INV !
```

```
! Definición de las ecuaciones del modelo!
```

```
INV >ECU 1, 'INV=INV(-1)+COMPRA-VENTA';  
INV >ECU 2, 'COMPRA=ENTREGA(INV(-1),REORDEN,PEDIDO,TIES)';  
INV >ECU 3, 'VENTA=DEMTOT+MIN(INV(-1)-DEMTOT,0)';  
INV >ECU 4, 'TIES=ARBOL(VTIE,PTIE,NT)';  
INV >ECU 5, 'DEMTOT=DEM+SPAT(-1)';  
INV >ECU 6, 'DEM=ARBOL(VDEM,PDEM,ND)';
```

```

INV >ECU 7,'SPAT=(SPAT(-1)+PEDAT)*MAY(OND,0)';
INV >ECU 8,'PEDAT=ORDEN(INV(-1),REORDEN,PAT,OND,TIES)';
INV >ECU 9,'DEMP=OND-PEDAT';
INV >ECU 10,'OND=(-MIN(INV(-1)-DEM,0))';
INV >ECU 11,'COSTOT=COSSUM+COSMAN';
INV >ECU 12,'COSCOM=VENTA*3+20*MAY(COMPRA,0)';
INV >ECU 13,'COSPER=3*DEMP';
INV >ECU 14,'COSMAN=MINV*5*TI+FER';
INV >ECU 15,'COSPA=10*MAY(PEDAT,0)';
INV >ECU 16,'MINV=(MINV(-1)*PER(-1)+INV)/PER';
INV >ECU 17,'COSSUM=COSSUM(-1)+COSPA+COSCOM+COSPER';
INV >TIE 2,102,5101; ! Definición del tiempo de simulación!
INV >CORR 100; ! El modelo será evaluado cien veces

! Definición de las variables de estadísticas; MEDCOS para
la media de
COSTOT y DESCOS para su desviación estándar !
INV >EST COSTOT,MEDCOS,DESCOS; INV >*
! Orden de ejecución de la simulación ! <COMPILANDO
ECUACIONES> <ORDENANDO ECUACIONES>

! Despliegue del diagnóstico en pantalla !
DIAGNOSTICO DE MONTE CARLO MODELO :INV
VARIABLES
  INV COMPRA VENTA TIES DEMENTOT
  DEM SPAT PEDAT OND DEMP
  COSTOT COSSUM COSMAN COSCOM COSPER
  MINV PER COSPA
PARAMETROS
  REORDEN PEDIDO VTIE PTIE NT
  VDEM PDEM ND PAT TI
ESTADISTICAS
  MEDIA DESVIACION ESTANDAR
  -----
  MEDCOS DESCOS
PERIODICIDAD : 2
INICIO : 102
FIN : 5101
NUMERO DE PERIODOS : 100
NUMERO DE CORRIDAS : 100
! Instrucciones del sistema operativo para compilar y correr
el programa generado por Monte Carlo !

```

```
>F4P SY1:INV = SY1:INV.PPR ,SY1:INV
>TKB SY1:INV/CP/FP=INV,MT/LB:MTC,MT/LB,C/LB:DKDK,C/LB
>RUN SY1:INV
! Instrucción que purga los archivos generados dejando solo
las últimas versiones !
>PIP SY1:INV ./PU
! Regreso al módulo de Monte Carlo !
>MTC INV
S.A.I. MODULO DE EJECUCION DE MONTECARLO VERSION 3.
! El nombre es mantenido y el sistema se encuentra en la
fase resultados ! ! Sólo es mantenida la variable COSTOT con
sus estadísticas. !
INV >MANTEN COSTOT; INV >* ! Orden de recuperar los
resultados ! INV >FIN ! Regreso al módulo de control de SAI
!
FUNCION >
```

B I B L I O G R A F I A

- 1.-
Akoff Russell Lincoln
Un Concepto de Planeación de Empresas
México, Limusa, 1972
- 2.-
Atkinson, A.,C.
A family of switching algorithms for the computer
generation of beta random variates
Biometrika 66(1) 1979,141-145.
- 3.-
Barraza Negrete David Alfonso
Blando Valenzuela Martha Eugenia
Gómez Tagle Bassoco Alfredo
Diseño y Desarrollo de un Sistema de Información
Computacional Interactivo.
Tesis Profesional
Universidad Iberoamericana México D.F. ,1980
- 4.-
Biernbaum Z. W.
Numerical Tabulation of the Distribution of Kolmogorov's
Statistic for the Finite Sample Size
JASA, 47, 1952, 425-441.
- 5.-
Bradly J. V.
Distribution-Free Statistical Tests
Prentice-Hall, Inc. New Jersey, 1968, 367-369.
- 6.-
Cheng, R.C.H.
Generating beta variates with nonintegral shape
parameters
Communications of the ACM 21(4) 1978, 317-322.
- 7.-
Conover W. J.
Practical Nonparametric Statistics
Jhon Wiley & Sons, 1971, 1980
- 8.-
Coss Bu Raúl
Simulación, Un Enfoque Práctico
Limusa, México.
- 9.-
Feller William.
On the Kolmogorov-Smirnov limit theorems for empirical
distributions
Annals of Mathematics Statistica, 177-189.

- 10.-
Fishman G. S.
Concepts and Methods in Discrete Event Digital
Simulation
Wiley, New York ,1973.
- 11.-
Freiberger W. Grenander U.
A Short Course In Computacional Probability and
Statistics Applied In Mathematics Sciences
Springer-Verlog 1971
- 12.-
García González Juan Carlos
Estructura Básica del Sistema de Analisis Interactivo
SAI
Tesis Profesional
Universidad Anahuac México D.F., 1983
- 13.-
Gnedenko
The Theory of Probability
Chelsa Publishing Co., New York, 1962, 384-401.
- 14.-
Hausrath Alfred H.
Venture Simulation in War, Buiness and Politica
Mc. Graw Hill, New York, 1971
- 15.-
Hernández Diego Bricio
Luis Javier Alvarez
Informe Monográfico: El Método de Monte Carlo
Departamento de Matemáticas Universidad Autónoma
Metropolitana - Iztapalapa
- 16.-
Hiller Frederick S., Liberman Gerald J.
Operations research
Holden-day
- 17.-
H.P. Williams
Model Building in Mathematical Programming
Chicester: Wiley, 1978
- 18.-
International Mathematical Statistical Libraries,INC.
THE IMSL LIBRARY
Volume 1 & 2
7500 Bellaire Boulevard Sixth Floor -GNB Building
Houston,Texas 77036 Telephone (713) 772 1927

- 19.-
Johns, M.D.
Erzeugung von Betaverteilten und Gammaverteilten
Zufallszahlen
Metrika 8(2) 1964, 5-15.
- 20.-
Knuth, Donald E.
The art of Computer Programming
Volume 2, Addison Wesley, 1969, 105 and 113.
- 21.-
Lewis, P.A.W., GOODMAN, A.S., AND MILLER, J.M.
PSEUDO-RANDOM GENERATOR FOR THE SYSTEM/360
IBM Systems Journal, 8(2), 1969, 136, 46.
- 22.-
Learmonth, G., P. and Lewis, P.A.W.
Naval Postgraduate School Random Number Generator
Package LLRANDOM
NPS55LW73061A Naval Postgraduate School, Monterey,
California, June, 1973.
- 23.-
Learmonth, G.P., and Lewis, P.A.W.
Statistical tests of some widely used and recently
proposed uniform random number generators
NPS55LW73111A, Naval Postgraduate School, Monterey,
California, november, 1973.
- 24.-
Lindgren B. W.
Statistical Theory
The MacMillan Company, New York.
- 25.-
Molino Enzo
Comunicación e Informática
Revista No. 2, Vol. 1, Diciembre 1980
- 26.-
Mood Alexander M., Franklin A. Graybill, Duane C. Boes
Introduction To The Theory Of Statistics
International Student Edition Mc. Graw Hill
- 27.-
Naylor T. J., J. L. Balintfy, D. S. Burdick, K. Chu
Computer Simulation Technique
Wiley New York, 1966.
- 28.-
Ostle Bernard
Statistics in Research
The Iowa state University Press

- 29.-
 PDP-11 FORTRAN Language Reference Manual
 Order No. DEC-11-LFLRA-C-D Maynard, Massachusetts , 1977
- 30.-
 Presidencia de la República, Dirección General del
 Secretariado Técnico de Gabinetes
 SISTEMA DE ANALISIS INTERACTIVO SAI MANUAL DEL USUARIO
 ISBN 968-809-750-0
- 31.-
 Relles, Daniel A.
 A simple algoritme for generating binomial random
 variables when n is large
 Journal of the American Statistical Association
 76,1972,612-613.
- 32.-
 Robert C. Meier, William T. Newell, Harold L. Pazer
 Técnicas de Simulación en Administración y Economía.
 México, Trillas, 1975.
- 33.-
 Robinson ,D.,W.,and Lewis, P.A.W.
 Generating Gamma and Cauchy random deviates
 An expression to the naval postgraduate school random
 number Package NPS-72 Rc
 75041,Naval Postgraduate School,Monterey, California,
 April 1975.
- 34.-
 Rubinstein Reuven Y.
 Simulation and the Monte Carlo Method
 Wiley Interscience
- 35.-
 Schaffer ,Henry E.
 Algoritm 369, generator of random numbers satisfying the
 Poisson distribution
 Comm. ACM,13(11)1970,49.
- 36.-
 Shapiro Samuel S. , Gross Alan J.
 Statistical Modeling Techniques
- 37.-
 Schmeiser,Bruce W. and Lal R.
 Squeeze methods for generating Gamma variates
 OREM 78009
 Southern Methodist University ,Dallas Texas,1979.

- 38.-
Schmeiser, Bruce W., and Babu, A.J.G.
Beta variate generation via exponential majorizing
functions
Operations Research 1980 (to appear)
- 39.-
Shan S. Kuo
Computer Application of Numerical Methods
Addison-Wesley, 1972
- 40.-
Snedecar George W., William G. Cochran
Statistical Methods
The Iowa state University Press
- 41.-
Snow, Richard H.
Algorithm 342, generator of random satisfying the
poisson distribution
comm. ACM, 11(12) 1968,
- 42.-
Springer, Herlihy, Beggs
Métodos Avanzados y Modelos
Serie de Matemáticas para la Dirección de Negocios
- 43.-
Warnier J. D. Flanagan B. M.
Programación Lógica
Tomo I Construcción de Programas
HONEYWELL BULL
Editores Técnicos Asociados S.A. Barcelona
- 44.-
Warnier J. D.
Programación Lógica
Tomo II Explotación de los Datos
HONEYWELL BULL
Editores Técnicos Asociados S.A. Barcelona
- 45.-
Yakowitz Sidney J.
Computational Probability Simulation
Addison-Wesley 1977