



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

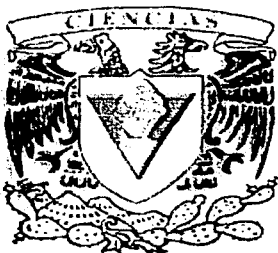
IMPLANTACIÓN DE LA FÓRMULA BDF DE ORDEN 1,
A PASO FIJO SIN CONTROL DE ERROR, PARA SIMULACIÓN
DE MODELOS MATEMÁTICOS DE PROCESOS DE PLANTAS
DE POTENCIA EN TIEMPO REAL

T E S I S
QUE PARA OBTENER EL TÍTULO DE
ACTUARIA

P R E S E N T A :
VIRGINIA GALLARDO FRANCO

**TESIS CON
FALLA DE ORIGEN**

DIRECTORES: M. en C. Rina Betzabeth Ojeda Castañeda
Dr. Jesús López Estrada



MÉXICO, D. F.



2002

**FACULTAD DE CIENCIAS
SECCION ESCOLAR**



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.

Paginación

Discontinua



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

División de Estudios Profesionales
Facultad de Ciencias, UNAM.


MAT. MARGARITA ELVIRA CHÁVEZ CANO
Jefa de la División de Estudios Profesionales de la
Facultad de Ciencias.
PRESENTE

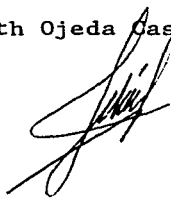
Comunicamos a usted que hemos revisado el trabajo de tesis que lleva el título:
"Implantación de la Fórmula BDF de Orden 1, a Paso Fijo sin Control de Error, para
Simulación de Modelos Matemáticos de Procesos de Plantas de Potencia en Tiempo Real"
realizado por: Virginia Gallardo Franco

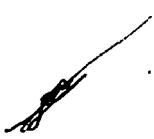
con número de cuenta 8629679-1, pasante de la carrera de Actuaría


Dicho trabajo cuenta con nuestro voto aprobatorio.

Atentamente

Director de Tesis M. en C. Rina Betzabeth Ojeda Castañeda 
Propietario

Propietario Dr. Jesús López Estrada 

Propietario Dr. Pablo Barrera Sánchez 

Suplente M. en C. María Elena García Alvarez 

Suplente M. en C. José López Estrada 

Consejo Departamental de Matemáticas
M. en A.P. María del Pilar Alonso Reyes
Coordinadora de la Carrera de Actuaría

CONSEJO DEPARTAMENTAL
DE
MATEMÁTICAS

AGRADECIMIENTOS

Agradezco a Dios por permitirme esta experiencia y como Dios esta en cada ser humano implícitamente las gracias van a mis Padres (Romeo y Estela), a mis Hermanos y a la Familia Parra (Julieta, Augusto, Gabriela, Dirce, Aarón, José Luis), a todos mis Maestros (en especial a la M. en C. Rina B. Ojeda Castañeda, al Dr. Jesús López Estrada, al Dr. Pablo Barrera y al M. en C. Gustavo Rodríguez Gómez), a todos aquellos que tuve la oportunidad de conocer en el IIE (en especial al Departamento de Simulación) y en la Facultad de Ciencias o en alguna otra parte debido a esta maravillosa experiencia. Dios los Bendiga a todos y nuevamente GRACIAS.

Virginia

**IMPLANTACIÓN DE LA FÓRMULA BDF DE ORDEN 1, A PASO FIJO
SIN CONTROL DE ERROR, PARA SIMULACIÓN DE MODELOS
MATEMÁTICOS DE PROCESOS DE PLANTAS DE POTENCIA EN
TIEMPO REAL**

VIRGINIA GALLARDO FRANCO

TABLA DE CONTENIDO

CAPÍTULO I. INTRODUCCIÓN.....	I-1
I.1 Supervisión de Plantas Generadoras de Energía Eléctrica.....	I-1
I.2 Simuladores de Plantas Generadoras de Energía Eléctrica.	I-2
I.2.1 Simuladores para entrenamiento de operadores	I-2
I.2.2 Simuladores para análisis y diseño	I-4
I.3 Sistemas de Simulación para Procesos Dinámicos.....	I-4
I.3.1 Sistemas basados en módulos	I-4
1.3.1.1 Métodos de solución desacoplada.....	I-6
1.3.1.2 Métodos de solución acoplada.....	I-8
I.3.2 Sistemas basados en ecuaciones.....	I-9
I.4 Generalidades de los Modelos del Simulador para Entrenamiento.....	I-10
I.4.1 Representación matemática (SEAD).....	I-11
I.4.2 Forma tradicional de solución	I-15
I.5 Tendencia en la Obtención de Modelos para Simuladores de Procesos Químicos para la Década del 2000	I-16
I.6 Objetivo de la Presente Tesis.....	I-18
I.7 Organización del Presente Trabajo	I-18
CAPÍTULO II. IMPLANTACIÓN DE LA FORMULA BDF DE ORDEN 1 PARA LA SOLUCIÓN DE SEAD's EN FORMA SIMULTÁNEA.....	II-1
II.1 Introducción	II-1
II.2 Métodos Multipaso Lineales	II-1
II.2.1 Las Fórmulas BDF.....	II-4
II.3 Aplicación de la Fórmula BDF de Orden 1 para la Solución de SEAD's en Forma Simultánea.....	II-7
II.3.1 Predicción del valor de inicio de la iteración	II-10
II.3.2 Algoritmo de la aplicación de la fórmula BDF de orden 1	II-12
II.4 Esquema Iterativo Newton para la Solución Simultánea.....	II-13
II.4.1 Aproximación numérica del Jacobiano JF	II-13
II.4.2 Solución del sistema lineal $JF\Delta y = -F$.....	II-14
II.4.3 Criterio de convergencia en las iteraciones del método de Newton.....	II-14
II.5 Programación del Algoritmo en Fortran	II-19
II.5.1 Definición de requerimientos.....	II-20
11.5.1.1 Planteamiento de objetivos.....	II-20
11.5.1.2 Análisis de objetivos	II-21
11.5.1.3 Especificación de requerimientos.....	II-24
II.5.2 Diseño.....	II-27
11.5.2.1 Arquitectura.....	II-27
11.5.2.2 Diagrama de estructura.....	II-30
11.5.2.3 Detalle de módulos del sistema.	II-30
11.5.2.3.1 Módulo 1: EHABDF.....	II-30
11.5.2.3.2 Módulo 1.1:SEAD.....	II-32
11.5.2.3.3 Módulo 1.2:DECOMP.....	II-33
11.5.2.3.4 Módulo 1.3:SOLVE.....	II-34
II.6 Validación Inicial del Algoritmo EHABDF	II-35
II.6.1 Modelos elegidos para la validación inicial.....	II-35
II.6.2 Pruebas realizadas	II-36
II.6.3 Análisis de los resultados.....	II-37
II.6.4 Resultados Modelo No. 1	II-38
11.6.4.1 Aproximación numérica.....	II-38
11.6.4.2 Costo computacional	II-41
II.6.5 Resultados Modelo No. 2	II-42
11.6.5.1 Aproximación numérica.....	II-42
11.6.5.2 Costo computacional	II-44

II.6.6 Resultados Modelo No. 3	II-45
II.6.6.1 Aproximación numérica	II-45
II.6.6.2 Costo computacional	II-49
II.6.7 Resumen de los resultados de la validación inicial	II-50

CAPÍTULO III. VALIDACIÓN DEL CÓDIGO EHABDF PARA LA SOLUCIÓN NUMÉRICA DEL SEAD EN TIEMPO REAL	III-1
III.1 Características Principales del Modelo Simplificado de la Planta	III-1
III.1.1 Sistema basado en módulos	III-1
III.1.2 Ecuaciones algebraico-diferenciales explícitas	III-4
III.1.2.1 Ecuaciones diferenciales	III-4
III.1.2.2 Ecuaciones algebraicas	III-5
III.1.2.3 Limitación de variables del SEAD	III-6
III.1.3 Secuencia de ejecución de las ecuaciones	III-8
III.1.3.1 Problemas encontrados en la secuencia del SEAD	III-8
III.1.3.2 Ordenamiento de la secuencia de ejecución de las ecuaciones	III-9
III.2 Pruebas Elegidas para la Validación del Código EHABDF	III-11
III.2.1 Prueba No 1. Modelo en estado estacionario	III-12
III.2.2 Prueba No. 2. Modelo en estado transitorio	III-12
III.2.3 Período de muestro de las variables del modelo	III-13
III.3 Generación de una Base de Referencia como Solución Exacta para la Aplicación de las Pruebas	III-13
III.4 Solución de las Ecuaciones Algebraico-Diferenciales del Modelo Ordenado por Medio de EHABDF	III-16
III.4.1 Adaptaciones realizadas al modelo	III-16
III.4.1.1 Ecuaciones diferenciales y algebraicas en forma implícita	III-16
III.4.1.2 Limitadores	III-17
III.4.1.3 Valores iniciales	III-18
III.4.2 Adaptaciones realizadas al código EHABDF para la aplicación de las pruebas	III-18
III.5 Resultados Obtenidos	III-18
III.6 Forma Alternativa de Solucionar el Modelo Ordenado por Medio de EHABDF	III-19
III.6.1 Adaptaciones realizadas al modelo para la aplicación de las pruebas	III-20
III.6.1.1 Ecuaciones diferenciales en forma implícita	III-20
III.6.1.2 Limitadores de variables de estado	III-20
III.6.2 Adaptaciones realizadas al código EHABDF para la aplicación de las pruebas	III-21
III.7 Resultados Obtenidos con la Forma Alternativa	III-21
III.7.1 Aproximación numérica	III-21
III.7.2 Costo computacional	III-27
III.7.3 Memoria requerida por el código EHABDF	III-31

CAPÍTULO IV. CONCLUSIONES	IV-1
IV.1 Mejoras al Código EHABDF	IV-3

APÉNDICE A	A-1
A.1 Secuencia de Ejecución de las Ecuaciones del Modelo Original	A-1
A.2 Ordenamiento del Modelo Original	A-11

APÉNDICE B. MATRIZ ESTRUCTURAL INICIAL	B-1
---	------------

APÉNDICE C	C-1
C.1 Ordenamiento de las Ecuaciones del Modelo	C-1

APÉNDICE D. MATRIZ ESTRUCTURAL FINAL, LEDET	D-1
--	------------

APÉNDICE E. MATRIZ ESTRUCTURAL FINAL, ORDENAMIENTO IMPLANTADO..... E-1

APÉNDICE F.....	F-1
F.1 Prueba de Consistencia Aplicada al Modelo Ordenado.....	F-1
F.1.1 Resultados de la prueba.....	F-2
F.2 Prueba de Estabilidad del Modelo Ordenado.....	F-2
F.2.1 Valores Característicos del Modelo.....	F-2
F.3 Solución y Validación del Modelo Ordenado.....	F-8
BIBLIOGRAFIA	G-1

CAPÍTULO I

INTRODUCCIÓN

I.1 Supervisión de Plantas Generadoras de Energía Eléctrica.

La supervisión y el control del funcionamiento de una planta generadora de energía eléctrica se realiza desde el cuarto de control de la planta. En el cuarto de control se encuentran tableros, computadoras, los operadores de la central e implícitamente los procesos que se están llevando a cabo en el funcionamiento de la planta.

Los tableros que se encuentran dentro del cuarto de control, contienen una serie de instrumentos que permiten observar el estado en que se encuentran funcionando los diferentes equipos involucrados en la generación de la energía eléctrica dentro de la planta, así encontramos por ejemplo, instrumentos que indican: temperaturas, presiones y vibraciones de los equipos que están trabajando dentro de la planta; en los tableros también se encuentran paneles de alarmas e instrumentos de control; todos éstos instrumentos sirven para que los operadores supervisen el adecuado funcionamiento de los equipos que conforman la planta, detecten condiciones anormales de operación y controlen así el estado de los procesos que se llevan a cabo en la obtención de la energía eléctrica dentro de la planta.

Las computadoras del cuarto de control sirven de interpretes entre los procesos que se están llevando a cabo en la planta y las acciones que realizan los operadores a través de los instrumentos de control de los tableros, es decir, la computadora recibe las señales que se generan en la planta, procesa la información y la manda a los tableros. Los operadores por su parte, atentos a la indicación de los diferentes instrumentos de los tableros, que les informan la evolución o el estado de los procesos que se están dando en la planta, toman las acciones debidas a través de los instrumentos de control de los tableros; éstas acciones realizadas se recogen nuevamente por la computadora, la cual procesa la información y la manda a los transductores de los equipos de la planta para que modifiquen el estado de operación de éstos.

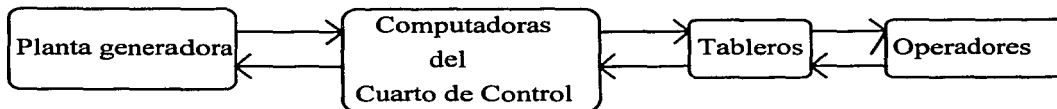


Fig. 1.1 esquema del "ciclo de control" de una planta generadora.

La figura 1.1 presenta la formación de un ciclo en la operación de la planta, este ciclo se conoce como "ciclo de control" en donde las acciones que tome el operador determina el buen o mal funcionamiento de la planta. Si el operador toma una acción equivocada, pueden resultar daños o desgastes a los equipos de la planta, y éstos generalmente son costosos.

Es importante entonces que los operadores estén debidamente capacitados y sepan que acciones de control tomar ante cualquier situación que pueda presentarse en los procesos de la central.

1.2 Simuladores de Plantas Generadoras de Energía Eléctrica.

1.2.1 Simuladores para entrenamiento de operadores

La capacitación de los operadores se lleva a cabo en los simuladores para entrenamiento, el objetivo de todos los simuladores para entrenamiento es reproducir fielmente el ambiente de trabajo de un operador con el fin de familiarizarlo con el funcionamiento del equipo de la planta y adiestrarlo en su operación a base de ejercicios repetidos. El entrenamiento se basa en estímulos visuales, auditivos y motrices, por medio de los cuales el estudiante adquiere una habilidad específica [Castelazo].

Un simulador para entrenamiento es una réplica total o parcial del cuarto de control de una planta, el cual "toma vida" gracias a que los procesos que se llevan a cabo en la central son sustituidos por modelos matemáticos representados con ecuaciones algebraicas y diferenciales, las cuales se solucionan en una (o varias) computadora(s). Los resultados obtenidos en la solución de las ecuaciones de los modelos son transmitidos a los tableros, éstos por su parte, indican al operador en adiestramiento el estado de los procesos simulados de la central, el operador entonces decide que acción tomar a través de los instrumentos de

control de los tableros, el tablero transmite a la computadora las acciones realizadas por el operador, la información generada por las acciones tomadas por el operador son registradas en los modelos matemáticos y éstos son nuevamente solucionados por la computadora, formándose así de nuevo un ciclo, al que denominaremos “ciclo simulador-operador”, en donde el proceso real ha sido sustituido por la computadora.

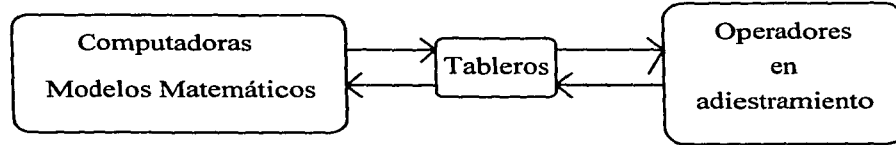


Fig. 1.2 “ciclo simulador-operador”.

De esta forma los operadores se capacitan o entrenan con muchas ventajas [Castelazo], algunas de las cuales son:

1. Se evita la utilización del equipo real de la planta.
2. Se pueden simular fallas en algún proceso de la planta.
3. Se evita el poner en riesgo la integridad física de los operadores.
4. Se puede comenzar, detener y continuar la simulación en cualquier momento durante la sesión de entrenamiento.
5. Se registra y facilita la evaluación de las acciones realizadas por los operadores.

Para que el adiestramiento tenga éxito es necesario que el ciclo del simulador sea muy similar al ciclo de operación de la planta real, es decir que:

- De acuerdo a las acciones que el operador tome en los tableros del simulador, los modelos matemáticos de los procesos de la planta, reflejen el comportamiento de los procesos reales de la planta bajo las acciones tomadas por el operador.
- El estado de los procesos simulados se observe en los tableros del simulador.
- El tiempo que le tome al simulador representar la evolución o estado de un proceso, sea el mismo tiempo que el proceso real tarda en evolucionar en la planta. Es decir, si se simula un proceso t unidades de tiempo, el simulador debe representar el estado o evolución alcanzado por el proceso en t unidades de tiempo, tardándose como máximo t unidades de tiempo, esto se conoce como simulación en tiempo real. Esta es una de las características más importantes de los simuladores para entrenamiento.

El tiempo real delimita en gran forma el desarrollo de los simuladores para adiestramiento de operadores, en particular el desarrollo de los modelos matemáticos que conforman al simulador, así como a los métodos numéricos que se utilizan para su resolución; ya que el tiempo de computación del simulador no debe de exceder el tiempo del período simulado [Rodríguez_1].

1.2.2 Simuladores para análisis y diseño

Además de los simuladores para adiestramiento existen simuladores para análisis y diseño de procesos de plantas generadoras. En éstos simuladores se analizan posibles diseños de futuras plantas o mejoras de las ya existentes, por lo cual los modelos matemáticos para éstos simuladores deben tener un alto grado de detalle, lo que implica el uso de métodos numéricos muy precisos para su solución, aún cuando no estén en tiempo real en la obtención de la solución, por lo tanto, lo más importante en estos simuladores es la exactitud en la aproximación de los resultados y no obtener la simulación en tiempo real.

1.3 Sistemas de Simulación para Procesos Dinámicos

Se ha visto como los procesos de la planta que se desean simular, son sustituidos por modelos matemáticos dentro del simulador, los modelos generalmente constan de un conjunto de ecuaciones algebraicas y diferenciales acopladas entre sí. En la ingeniería química se han desarrollado dos técnicas para la obtención de los modelos conocidas como [Cameron]:

- Sistemas basados en módulos
- Sistemas basados en ecuaciones

1.3.1 Sistemas basados en módulos

En los sistemas dinámicos modulares cada unidad o equipo físico contenido dentro de un proceso dado, por ejemplo: reactor, válvula, caldera, turbina, etc., es un módulo auto contenido descrito por ecuaciones diferenciales y ecuaciones algebraicas generalmente acopladas entre sí. El proceso se construye a partir de éstos módulos los cuales son ligados entre sí de acuerdo a la topología del proceso.

Un programa ejecutivo coordina el llamado de los módulos en una secuencia predeterminada, proporcionando las variables de entrada a cada módulo y recibiendo los valores de todas las variables al final del intervalo de tiempo. El intervalo de tiempo es entonces nuevamente avanzado y la secuencia de llamado a los módulos es repetida. Este procedimiento se desarrolla hasta que el tiempo final es alcanzando.

Para el programa ejecutivo, los módulos son "cajas negras" las cuales generan salidas para un conjunto dado de entradas.

Generalmente los módulos son subrutinas altamente estructuradas, las cuales además de contener las ecuaciones relevantes a la unidad, también contienen una secuencia computacional de las ecuaciones. Cuando los módulos contienen ecuaciones algebraicas que deben resolverse en forma simultánea, éstas se resuelven a través de métodos numéricos apropiados para ello, por ejemplo el método de Newton-Raphson. La figura 1.3.1 muestra una estructura modular típica.

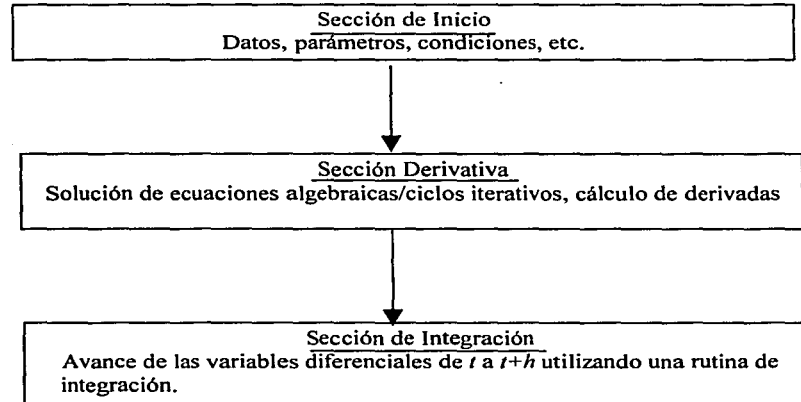


Fig. 1.3.1 Estructura típica de un módulo en sistemas de simulación modular.

Una de las consecuencias de la aproximación modular es que el usuario y/o autor debe tener gran cuidado al estructurar las ecuaciones algebraicas en una correcta secuencia de computo así como al seleccionar métodos apropiados para resolver los sistemas de ecuaciones algebraicas que deban resolverse en forma simultánea. Esto puede causar problemas para el usuario general, el cual no quiere verse involucrado con este tipo de decisiones cuando un nuevo módulo necesite ser añadido al sistema [Cameron, Ojeda_1]

Uno de los problemas fundamentales de los sistemas basados en módulos involucra el grado de acoplamiento que puede existir entre las ecuaciones de los diferentes módulos. Esto lleva a dos divisiones básicas dentro de las principales ideas de la simulación modular [Cameron, Ojeda_1].

- Métodos de solución desacoplada
- Métodos de solución acoplada

1.3.1.1 Métodos de solución desacoplada

La solución de las ecuaciones en un módulo particular es calculada independientemente del resto de los módulos. Para obtener la solución del sistema completo en Δt unidades de tiempo de simulación, el programa ejecutivo divide éstas Δt unidades de tiempo en tantos intervalos de tiempo como módulos tengan que resolverse, llama en una secuencia determinada a cada módulo, el módulo llamado resuelve las ecuaciones algebraicas y diferenciales contenidas en él, utilizando como paso de integración la longitud del intervalo correspondiente al módulo, una vez obtenida la solución de las ecuaciones en el módulo, se obtienen las salidas de éste y son proporcionadas al siguiente módulo que se llama, este proceso se sigue hasta finalizar la solución de todos los módulos, el control de la ejecución es entonces regresado al programa ejecutivo. El procedimiento es nuevamente repetido hasta que el tiempo final de simulación ha sido alcanzado. Esto se ilustra esquemáticamente en la figura 1.3.1.1

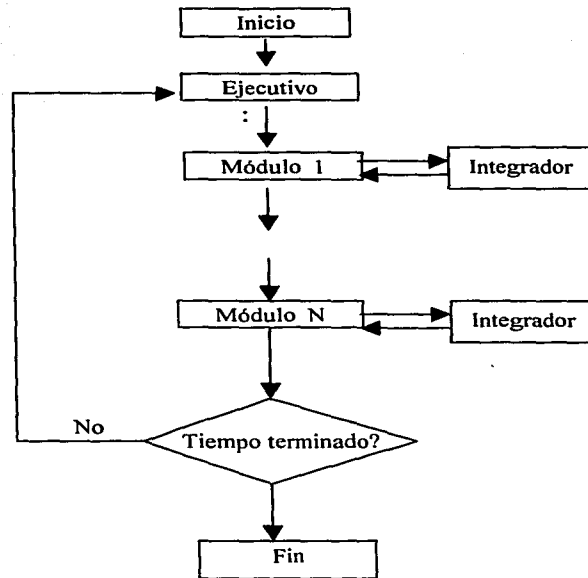


Fig. 1.3.1.1 Secuencia de computación en una Simulación Modular Desacoplada

Como puede verse el acoplamiento de todas las ecuaciones ocurre al final de un intervalo de simulación y no en cada paso de integración en que se resuelven las ecuaciones diferenciales, por lo que la exactitud de la solución puede no ser muy buena, a menos que los intervalos en que se resuelven las ecuaciones diferenciales sean muy pequeños[Cameron].

Generalmente los métodos desacoplados son utilizados en la simulación de sistemas muy grandes. Debido a que cada módulo llama internamente al método integrador, el almacenamiento de información requerido por el método integrador se determina por el módulo con el mayor número de ecuaciones diferenciales. Lo cual permite realizar simulaciones de sistemas muy grandes, además permite seleccionar el tipo de método integrador más apropiado a cada módulo, dependiendo de la naturaleza de las ecuaciones presentes en el módulo[Cameron].

1.3.1.2 Métodos de solución acoplada

Para garantizar una mayor exactitud en la simulación dinámica, se emplea el acoplamiento de todas las ecuaciones diferenciales presentes en los módulos en cada paso de integración que se avanza la solución de todo el sistema. De esta forma, todas las ecuaciones diferenciales son integradas juntas en el mismo intervalo de tiempo. El procedimiento básico se muestra en la figura 1.3.1.2 [Cameron].

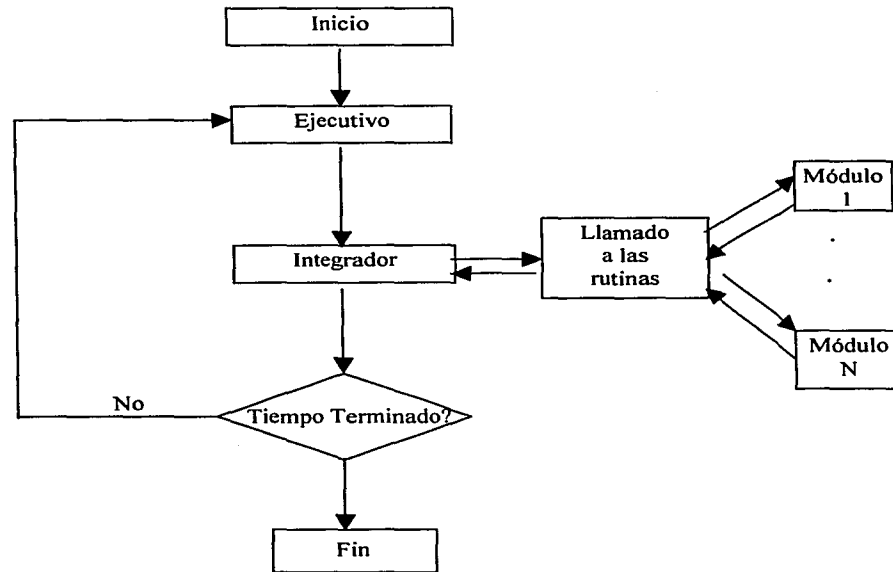


Fig. 1.3.1.2 Secuencia de computación en una Simulación Modular Acoplada

La adopción de esta estrategia significa un incremento considerable en los requerimientos de almacenamiento del método integrador, debido a que se requiere almacenar todas las variables diferenciales de todos los módulos. Esto puede significar una limitación en el número de unidades de proceso que puedan ser simuladas dependiendo de la computadora que vaya a ser utilizada.

1.3.2 Sistemas basados en ecuaciones

En contraste con el sistema modular donde los módulos son llamados en una secuencia predeterminada para generar los valores de entrada y de salida para cada módulo, el sistema orientado en ecuaciones trabaja directamente con todo el conjunto de ecuaciones [Cameron].

La forma de obtener la solución de la simulación en los sistemas basados en ecuaciones es fundamentalmente diferente al de la solución modular en que la estructura básica de todo el sistema de ecuaciones es primeramente analizada, por medio de la representación del sistema de ecuaciones a través de una gráfica dirigida, que muestra las relaciones de dependencia funcional entre todas las variables del sistema (por ejemplo flujos, temperaturas, y presiones involucrados en los procesos, así como parámetros de los equipos, etc.). Esta gráfica dirigida es analizada automáticamente para producir una secuencia computacional ordenada y eficiente. Los principales aspectos de este procedimiento se muestran en la siguiente figura 1.3.2 [Cameron].

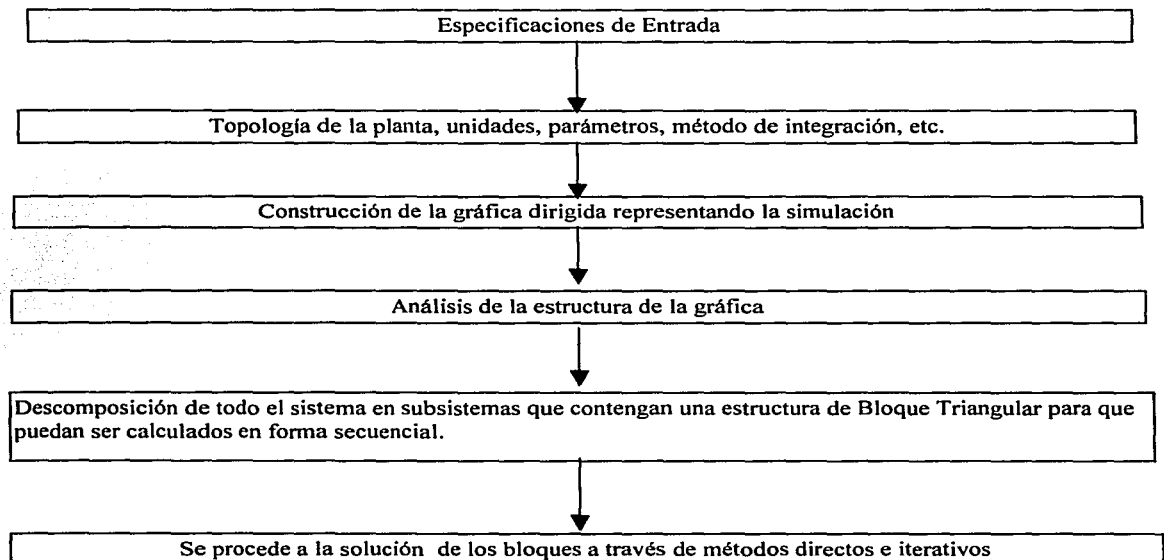


Fig. 1.3.2 Estructura de los Sistemas de Simulación Basados en Ecuaciones

Las ecuaciones diferenciales ordinarias presentes en el sistema son resueltas usando esquemas de integración simple (por ejemplo métodos explícitos como el Euler o los métodos Runge-Kutta explícitos ó un método implícito como el método de Euler hacia atrás). Los métodos explícitos producen relaciones algebraicas las cuales pueden ser resueltas directamente. Los métodos implícitos producen relaciones algebraicas las cuales requieren de iteraciones para su solución. Así la simulación completa (tanto las ecuaciones algebraicas como las diferenciales) consiste de un conjunto de ecuaciones algebraicas lineales y no lineales las cuales son resueltas en cada paso de integración [Cameron]. Para resolver el sistema de ecuaciones no lineales resultante, se utilizan métodos iterativos como el método de Newton, el cual requiere del jacobiano del sistema de ecuaciones no lineales, así que si el sistema es muy grande se requiere de una gran cantidad de almacenamiento de información, si el sistema tiene una estructura rala se pueden explotar las técnicas de matrices ralas para el manejo de este jacobiano.

Esta aproximación tiene la ventaja sobre la modular, en que al trabajar directamente con todo el conjunto de ecuaciones, permite una mayor flexibilidad en la elección de variables de diseño del simulador, pues no importa la dirección del flujo de información de éstas variables como en el caso de la aproximación modular. Sin embargo, el conocimiento a priori de todo el conjunto de ecuaciones requeridas para los procesos que se van a simular hace que los sistemas basados en ecuaciones sean restringidos a usuarios expertos [Cameron].

1.4 Generalidades de los Modelos del Simulador para Entrenamiento

Las tres características citadas en la sección I.2.1 (tiempo real, cierto grado de aproximación con la realidad del fenómeno simulado y la interacción mutua entre el operador y el simulador) que deben cumplir los simuladores para entrenamiento, son las directrices que marcan el desarrollo de los simuladores para entrenamiento, en particular del desarrollo de los modelos matemáticos y de los métodos numéricos que se emplean para resolverlos. Por estas razones los modelos matemáticos son desarrollados con muchas simplificaciones, hasta lograr cierto grado de aproximación con la realidad. Asimismo se requiere de métodos numéricos que obtengan la solución de los modelos en el menor tiempo posible tomando en cuenta las

características de los modelos. La forma tradicional en que se han desarrollado los modelos matemáticos de los simuladores para adiestramiento de operadores de plantas generadoras de energía eléctrica, es a través del desarrollo de sistemas basados en módulos con un método de solución desacoplada (ver I.3.1 y I.3.1.1). De esta manera cada módulo describe mediante un conjunto de ecuaciones diferenciales ordinarias y un conjunto de ecuaciones algebraicas una parte de todo el modelo, físicamente el módulo representa una sección de la planta, por ejemplo: la caldera, la turbina, etc. [Ojeda_1].

I.4.1 Representación matemática (SEAD)

La formulación de los modelos matemáticos que componen a un simulador para adiestramiento, se elabora a través de las leyes de la conservación de masa, energía y momentum [Luyben].

La aplicación de las leyes de la conservación de masa, momentum y energía a los procesos pertenecientes a una planta generadora de energía eléctrica, da como resultado un conjunto de ecuaciones algebraicas acopladas a un conjunto de ecuaciones diferenciales ordinarias, a este conjunto de ecuaciones se le conoce como sistema de ecuaciones algebraico-diferenciales [Cameron, Ojeda_1] y nos referiremos a él simplemente como SEAD.

La representación matemática que tradicionalmente se ha obtenido para el SEAD en los procesos de plantas generadoras se presenta a continuación:

$$\begin{array}{rcl} \mathbf{y}' = \mathbf{f}(\mathbf{y}, \mathbf{z}, t) & (1.4.1.1) & \\ \mathbf{0} = \mathbf{g}(\mathbf{y}, \mathbf{z}, t) & (1.4.1.2) & \} \quad (1.4.1) \end{array}$$

Donde

\mathbf{y} el vector de variables diferenciales (variables de estado), $\mathbf{y} \in \mathbb{R}^m$

\mathbf{y}' el vector de derivadas del vector \mathbf{y} , con $\mathbf{y}' \in \mathbb{R}^m$

\mathbf{z} el vector de variables algebraicas, con $\mathbf{z} \in \mathbb{R}^n$

t la variable independiente tiempo, $t \in [t_0, t_N]$.

$\mathbf{f} : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m$

$\mathbf{g} : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$

Siempre se dan condiciones de inicio para las variables diferenciales como $\mathbf{y}(t_0)=\mathbf{y}_0$, y valores iniciales al tiempo t_0 para las variables algebraicas del sistema no lineal, es necesario calcular los valores de las variables algebraicas lineales en el punto inicial t_0 .

A este tipo de SEAD se le conoce como un SEAD semiexplícito [Brenan] y es un caso particular de la forma general de un SEAD el cual se define como un conjunto de ecuaciones diferenciales ordinarias implícitas

$$\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0} \quad (1.4.2)$$

donde

$$\mathbf{F} : \Omega \subset \mathbb{R}^k \times \mathbb{R}^k \times \mathbb{R} \rightarrow \mathbb{R}^k$$

y la matriz jacobiana parcial de \mathbf{F} respecto a \mathbf{x}' ($\partial \mathbf{F} / \partial \mathbf{x}'$) es singular en todos los valores de sus argumentos [Roswitha]. En efecto, si se toma $\mathbf{x} = (\mathbf{y}, \mathbf{z})^T$ donde T indica el operador transpuesto y se define \mathbf{F} como

$$\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \begin{pmatrix} \mathbf{y}' - \mathbf{f}(\mathbf{y}, \mathbf{z}, t) \\ \mathbf{g}(\mathbf{y}, \mathbf{z}, t) \end{pmatrix} \quad (1.4.2.1)$$

en este caso $k = m+n$ (m variables de estado, n variables algebraicas), entonces es directo ver que

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}'} = \begin{pmatrix} \mathbf{I}_{m \times m} & \mathbf{0}_{m \times n} \\ \mathbf{0}_{n \times m} & \mathbf{0}_{n \times n} \end{pmatrix}$$

Esto es, que $\partial \mathbf{F} / \partial \mathbf{x}'$ es singular.

Una propiedad de los SEAD's conocida como el índice del SEAD, tiene un papel muy importante en su clasificación y comportamiento. Para motivar la definición de índice Brenan [Brenan] hace considerar el sistema semiexplícito (1.4.1)

$$\begin{aligned} \mathbf{y}' &= \mathbf{f}(\mathbf{y}, \mathbf{z}, t) & (1.4.1.1) \\ \mathbf{0} &= \mathbf{g}(\mathbf{y}, \mathbf{z}, t) & (1.4.1.2) \end{aligned} \quad \left. \vphantom{\begin{aligned} \mathbf{y}' &= \mathbf{f}(\mathbf{y}, \mathbf{z}, t) \\ \mathbf{0} &= \mathbf{g}(\mathbf{y}, \mathbf{z}, t) \end{aligned}} \right\} (1.4.1)$$

y hace notar que si se hace la diferenciación del sistema (1.4.1.2) con respecto a t , se obtiene

$$\begin{aligned} \mathbf{y}' &= \mathbf{f}(\mathbf{y}, \mathbf{z}, t) & (1.4.3.1) \\ (\partial \mathbf{g} / \partial \mathbf{y}) \mathbf{y}' + (\partial \mathbf{g} / \partial \mathbf{z}) \mathbf{z}' &= -(\partial \mathbf{g} / \partial t) & (1.4.3.2) \end{aligned} \quad \left. \vphantom{\begin{aligned} \mathbf{y}' &= \mathbf{f}(\mathbf{y}, \mathbf{z}, t) \\ (\partial \mathbf{g} / \partial \mathbf{y}) \mathbf{y}' + (\partial \mathbf{g} / \partial \mathbf{z}) \mathbf{z}' &= -(\partial \mathbf{g} / \partial t) \end{aligned}} \right\} (1.4.3)$$

Si $(\partial g/\partial z)$ es no singular, entonces se obtiene en (1.4.3) un conjunto de ecuaciones diferenciales ordinarias implícitas y se dice que el sistema (1.4.3) tiene índice 1. Si este no es el caso, entonces se supone que con manipulación algebraica y cambios de coordenadas se puede volver a escribir (1.4.3) como (1.4.1) pero con diferentes y y z . Una vez más se diferencia el sistema algebraico (1.4.1.2) con respecto a t . Si de esta diferenciación se puede obtener un sistema de ecuaciones diferenciales implícitas, se dice que el problema original tiene índice 2. Si este no es el caso, se repite el proceso. El número de pasos de diferenciación requeridos en este procedimiento es el índice del SEAD [Brenan].

A continuación se presenta la definición de índice [Brenan].

Definición 1.4.1 El mínimo número de veces que todo el sistema o parte del sistema (1.4.2) debe ser diferenciado con respecto a t para poder determinar x' como una función de x y t , es el índice del SEAD (1.4.2).

Para asegurar la existencia de una solución única del problema general semiexplícito (1.4.1) se asume que f es continua en todos sus argumentos y de Lipschitz respecto a y y z , que g es continua en todos sus argumentos y diferenciable en z y que el Jacobiano del sistema algebraico respecto a las variables algebraicas $(\partial g/\partial z)$ sea no singular [Cameron], es decir que el SEAD tenga índice 1.

A continuación se presenta el teorema 1.4.1 que garantiza la existencia y unicidad de la solución de los SEAD's de la forma (1.4.1) semi-implícitos de índice uno [López].

Teorema 1.4.1 Si f es continua en todos sus argumentos y de Lipschitz (localmente) respecto a y y z , y g es continua en todos sus argumentos y diferenciable en z y $\partial g/\partial z$ es no singular entonces el SEAD

$$\left. \begin{aligned} y' &= f(y, z, t) \\ 0 &= g(y, z, t) \end{aligned} \right\} \quad (1.4.4.1)$$

con condiciones iniciales consistentes $\mathbf{y}(t_0) = \mathbf{y}_0$, $\mathbf{g}(\mathbf{y}_0, \mathbf{z}_0, t_0) = \mathbf{0}$, tiene (localmente) una única solución.

Demostración:

Si $\partial \mathbf{g} / \partial \mathbf{z}$ es no singular, entonces por el teorema de la función implícita [Courant], en una vecindad del punto (\mathbf{y}_0, t_0) denotada como $V(\mathbf{y}_0, t_0)$ existe una función diferenciable φ tal que

a) $\mathbf{z} = \varphi(\mathbf{y}, t)$

b) $\mathbf{0} = \mathbf{g}(\mathbf{y}, \varphi(\mathbf{y}, t), t)$, para todos los puntos (\mathbf{y}, t) que se encuentran dentro de la vecindad de (\mathbf{y}_0, t_0) .

Luego sustituyendo \mathbf{z} por $\varphi(\mathbf{y}, t)$ en la primera expresión de (1.4.4.1), se tiene que

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, \varphi(\mathbf{y}, t), t) \quad (1.4.4.2)$$

Sea ahora

$$\mathbf{h}(\mathbf{y}, t) \equiv \mathbf{f}(\mathbf{y}, \varphi(\mathbf{y}, t), t) \quad (1.4.4.3)$$

y utilizando (1.4.4.3) el sistema de ecuaciones diferenciales ordinarias (1.4.4.2) se puede escribir como

$$\mathbf{y}' = \mathbf{h}(\mathbf{y}, t) \quad (1.4.4.4)$$

con valores iniciales (\mathbf{y}_0, t_0) tales que $\mathbf{y}(t_0) = \mathbf{y}_0$, $\mathbf{g}(\mathbf{y}_0, \varphi(\mathbf{y}_0, t_0), t_0) = \mathbf{0}$

El cual tiene una única solución en una región D , donde D esta definida como:

$a \leq t \leq b$, $-\infty \leq y_j \leq \infty$, donde las y_j con $j=1, \dots, m$ son las componentes del vector \mathbf{y} y a y b finitos, la función \mathbf{h} es continua en todos los puntos $(\mathbf{y}, t) \in D$ por ser una composición de funciones continuas.

Ahora, sean (\mathbf{y}, t) y $(\mathbf{y}^*, t) \in V(\mathbf{y}_0, t_0)$, por ser \mathbf{f} localmente de Lipschitz

$$\|\mathbf{h}(\mathbf{y}, t) - \mathbf{h}(\mathbf{y}^*, t)\| = \|\mathbf{f}(\mathbf{y}, \varphi(\mathbf{y}, t), t) - \mathbf{f}(\mathbf{y}^*, \varphi(\mathbf{y}^*, t), t)\| \leq L_f (\|\mathbf{y} - \mathbf{y}^*\| + \|\varphi(\mathbf{y}, t) - \varphi(\mathbf{y}^*, t)\|) \quad (a)$$

Y como $\varphi \in C^1$, existe L_φ tal que

$$\|\varphi(\mathbf{y}, t) - \varphi(\mathbf{y}^*, t)\| \leq L_\varphi \|\mathbf{y} - \mathbf{y}^*\| \quad (b)$$

De (b) en (a), se obtiene que

$$\|\mathbf{h}(\mathbf{y}, t) - \mathbf{h}(\mathbf{y}^*, t)\| \leq L_f (\|\mathbf{y} - \mathbf{y}^*\| + L_\varphi \|\mathbf{y} - \mathbf{y}^*\|) \leq L \|\mathbf{y} - \mathbf{y}^*\|$$

donde $L = L_f + L_f L_\varphi$.

Lo que demuestra que la función h es de Lipschitz respecto a y para todos los puntos (y, t) y $(y^*, t) \in V(y_0, t_0)$. \square

1.4.2 Forma tradicional de solución

La solución de los modelos en un simulador para entrenamiento se obtiene en un conjunto discreto de puntos equidistantes entre sí: t_0, t_1, \dots, t_N , donde $t_{i+1} - t_i = h$ para $i = 0, \dots, N$, h se conoce como paso de integración. Para avanzar la solución del modelo (1.4.1), del tiempo t_0 al tiempo t_1 se realiza lo siguiente:

Dados los valores iniciales de las variables de estado y_0 y de las variables algebraicas z_0 , así como el valor del tiempo inicial t_0 , se itera el conjunto de ecuaciones algebraicas (1.4.1.2) un número fijo de veces, permaneciendo y_0 constante a lo largo de las iteraciones y se obtiene de esta forma un z_1 . Una vez obtenido el vector de variables algebraicas z_1 , este se sustituye en el conjunto de ecuaciones diferenciales (1.4.1.1) quedando como $y' = f(y_0, z_1, t_0)$, el cual es entonces integrado. El ciclo anterior es repetido por paso de integración, hasta alcanzar el tiempo final de la simulación [Ojeda_1, Rodríguez_1, López_1].

Es importante mencionar que el Tiempo de Unidad de Procesamiento Central por cada paso de integración ($TUPC_h$) que tome la computadora en la solución del modelo, debe ser como máximo el paso de integración h , es decir:

$$TUPC_h \leq h$$

para estar dentro de una simulación en tiempo real. También recordemos que la solución de los modelos se realiza tomando en cuenta las acciones tomadas en los instrumentos de control realizadas por los operadores en entrenamiento, debido a esto, los métodos integradores que se utilizan para la solución de (1.4.1.1) se implementan a paso y orden fijo, sin control de error [Rodríguez_1], los métodos integradores utilizados tradicionalmente son el Euler explícito o los métodos de Runge-Kutta, así como los métodos multitasa.

I.5 Tendencia en la Obtención de Modelos para Simuladores de Procesos Químicos para la Década del 2000.

Actualmente los avances logrados en la industria de la computación dentro de las áreas de *hardware* y de *software* permiten la existencia de computadoras con una gran capacidad de almacenamiento y una mayor rapidez en el procesamiento de cálculos numéricos, esto permite que las nuevas tendencias en la obtención de los modelos para simuladores de procesos químicos apunten a combinar las técnicas utilizadas para la obtención de los modelos modulares y las técnicas para la obtención de modelos basados en ecuaciones. De esta forma se pueden obtener las ventajas de ambas técnicas [Boston]. Una de las principales ventajas es la formulación individual de cada uno de los componentes o unidades físicas a simular, donde el conjunto de ecuaciones algebraico-diferenciales que representa a cada uno de estos módulos se obtienen como ecuaciones implícitas de la forma:

$$f_i(y_i', y_i, z_i, t) = 0 \quad (1.5.1.1)$$

$$g_i(y_i, z_i, t) = 0 \quad (1.5.1.2)$$

Donde:

- i es la componente i del sistema a simular
- y_i el vector de variables diferenciales (variables de estado), que se encuentran dentro de la formulación de la componente i -ésima
- y_i' el vector de derivadas del vector y_i , que se encuentran dentro de la formulación de la componente i -ésima
- z_i el vector de variables algebraicas que se encuentran dentro de la formulación de la componente i -ésima
- t la variable independiente tiempo, $t \in [t_0, t_N]$.

esto permite la unión de todas las ecuaciones algebraico-diferenciales presentes en cada módulo i , dando por resultado un conjunto final de ecuaciones algebraico-diferencial en forma implícita, representado como:

$$f(y', y, z, t) = 0 \quad (1.5.2.1)$$

$$g(y, z, t) = 0 \quad (1.5.2.2)$$

} (1.5.2)

Donde:

y el vector de variables diferenciales (variables de estado de todo el simulador), $y \in \mathbb{R}^m$

y' el vector de derivadas del vector y , con $y' \in \mathbb{R}^m$

z el vector de variables algebraicas de todo el simulador, con $z \in \mathbb{R}^n$

t la variable independiente tiempo, $t \in [t_0, t_N]$.

$$f : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^m$$

$$g : \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$$

El cual debe resolverse en forma simultánea para el conjunto de variables algebraicas z y para el conjunto de variables diferenciales y .

En el caso del modelo completo del simulador de una planta de potencia, el SEAD que se obtiene por lo general es lineal en las variables diferenciales:

$$\begin{aligned} y' - f(y, z, t) &= 0 \\ g(y, z, t) &= 0 \end{aligned} \quad (1.5.3)$$

donde los componentes de este sistema se definen en el sistema (1.4.1), además se requieren valores iniciales consistentes al tiempo t_0 de y_0 , z_0 e y_0' , es decir tales que:

$$\begin{aligned} y_0' - f(y_0, z_0, t_0) &= 0 \\ g(y_0, z_0, t_0) &= 0 \end{aligned}$$

Gear fue el pionero en resolver este tipo de SEAD's, a través de la implantación de las fórmulas BDF, utilizando paso de integración y orden variable dentro del código DIFSUB [Gear_2].

I.6 Objetivo de la Presente Tesis

El objetivo de la presente tesis es validar y documentar un programa de cómputo denominado EHABDF[Rodríguez_4] que contiene la implantación de la fórmula BDF de orden uno a paso fijo, sin control de error, desarrollado en el Departamento de Simulación del IIE para obtener en tiempo real la solución numérica en forma simultánea de los SEAD's que conforman los modelos de procesos de plantas generadoras de energía eléctrica. La validación del programa se realizará resolviendo un modelo simplificado de una planta de potencia.

I.7 Organización del Presente Trabajo

En el capítulo II se introduce el marco teórico de la implantación de la fórmula BDF de orden 1 a paso fijo para la obtención de la solución en forma simultánea de un sistema de ecuaciones algebraico-diferenciales, y se presentan los resultados de la primera validación del código EHABDF con ejemplos teóricos de modelos muy sencillos.

En el capítulo III se hace la validación del código EHABDF con el modelo simplificado de una planta de potencia, en el capítulo IV se dan las conclusiones y recomendaciones del presente trabajo.

Finalmente en los apéndices A, B, C, D, E y F se presentan los resultados obtenidos en el ordenamiento del SEAD del modelo de la planta de potencia adaptado para la validación del código EHABDF.

CAPÍTULO II

IMPLANTACIÓN DE LA FORMULA BDF DE ORDEN 1 PARA LA SOLUCIÓN DE SEAD's EN FORMA SIMULTÁNEA

II.1 Introducción

En el presente capítulo se describen los detalles involucrados en la implantación de la fórmula BDF de orden 1 a paso fijo sin control de error para la solución en forma simultánea de un SEAD de la forma (1.5.3) definido en el capítulo anterior, con índice 1 y que cumple con las hipótesis del teorema (1.4.1) y que además tiene valores iniciales consistentes al tiempo t_0 de y_0 , z_0 e y_0' siguiendo la metodología planteada por Gear en [Gear_2].

II.2 Métodos Multipaso Lineales

Las fórmulas BDF forman parte de los métodos multipaso lineales utilizados para resolver numéricamente ecuaciones diferenciales ordinarias con valores iniciales:

$$y' = f(y, t), y(t_0) = \eta \quad (2.2.1)$$

La obtención de la solución de la ecuación diferencial a través de estos métodos, es a partir de la discretización del intervalo de tiempo donde se requiere la solución, de esta forma si se busca el valor de la solución en el intervalo $a \leq t \leq b$ a y b finitos, la solución se obtendrá únicamente en la secuencia de puntos de tiempo $\{t_n\}$ definida como $t_n = t_0 + nh$, $n=0,1,2,\dots$, $(b-a)/h$, h es conocido como el paso de integración. Así para cada t_n se obtiene una aproximación y_n a la solución exacta $y(t_n)$ de la ecuación diferencial [Lambert].

La forma para obtener la secuencia $\{y_{n+k}\}$ de la solución exacta $\{y(t_{n+k})\}$ en los métodos multipasos lineales es a través de una relación lineal entre y_{n+j} , y f_{n+j} dada como:

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f_{n+j} \quad (2.2.2)$$

donde:

k es el número de pasos

$f_{n+j} = f(y_{n+j}, t_{n+j})$ con $j=0,1,\dots,k$

h es el paso de integración.

Los coeficientes α_j y β_k con $j = 0, 1, \dots, k$ son constantes y se asume que $\alpha_k \neq 0$ y que al menos unos de los coeficientes α_0 y β_0 es diferente de cero. Si $\beta_k = 0$ se dice que el método es explícito, si $\beta_k \neq 0$ se dice que el método es implícito. La implantación de alguno de estos métodos requiere de varias cosas, entre ellas la elección del paso de integración $h > 0$ y fijo, de forma tal que el error global en la obtención de la solución ($e_{n+j} = y(t_{n+j}) - y_{n+j}$) permanezca bajo control cuando $n \rightarrow \infty$ [Lambert]. El criterio para la elección de h se obtiene a través de la Teoría de Estabilidad Débil [Lambert], de donde se obtiene lo que se conoce como polinomio característico o polinomio de estabilidad del método, el cual se define a continuación:

$$\Pi(r, \bar{h} = h\lambda) = \sum_{j=0}^k (\alpha_j - h\lambda\beta_j)r^j = \sum_{j=0}^k \alpha_j r^j - h\lambda \sum_{j=0}^k \beta_j r^j = \rho(r) - \bar{h} \sigma(r) = 0 \quad (2.2.3)$$

Donde:

Los coeficientes α_j y β_k con $j = 0, 1, \dots, k$ son los coeficientes originales del método,

r es raíz del polinomio,

λ es el valor de $\frac{\partial f}{\partial y}$,

$\bar{h} = h\lambda$,

$$\rho(r) = \sum_{j=0}^k \alpha_j r^j$$

$$\sigma(r) = \sum_{j=0}^k \beta_j r^j$$

Definición 2.2.1 El método multipaso lineal (2.2.2) se dice que es absolutamente estable para un \bar{h} sí, para ese \bar{h} , todas las raíces r_s del polinomio de estabilidad (2.2.3) satisfacen que $|r_s| < 1$, $s=1, 2, \dots, k$, y es absolutamente inestable para \bar{h} cuando ocurra cualquier otra cosa. Un intervalo (α, β) de la recta real se dice que es un intervalo de estabilidad absoluta si el método es absolutamente estable para toda $\bar{h} \in (\alpha, \beta)$. Si el método es absolutamente inestable para toda \bar{h} se dice que no tiene intervalo de estabilidad absoluta.

El intervalo de estabilidad absoluta es determinado únicamente por los coeficientes del método. Sin embargo, el valor más grande correspondiente de h para el cual el error global no se incrementará es, como puede verse en $\bar{h} = h\lambda$, dependiente de λ y así de la ecuación diferencial particular cuya solución se está buscando. Para una ecuación no lineal, $\frac{\partial f}{\partial y}$ no será constante, y se debe elegir una cota para λ o un valor típico de la misma, posiblemente que permanezca sobre un subintervalo del rango de integración. Si un valor diferente para λ se elige en un subsecuente subintervalo, habrá un correspondiente cambio en el valor máximo permitido para h .

Los métodos multipasos lineales presentados en (2.2.2) pueden ser implementados también para resolver sistemas de ecuaciones diferenciales ordinarias (SEDO's) con valores iniciales:

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, t), \mathbf{y}(t_0) = \boldsymbol{\eta} \quad (2.2.4)$$

donde

\mathbf{y} es el vector de variables de estado y $\mathbf{y} \in \mathfrak{R}^m$

$$\mathbf{f} = (f_1, f_2, \dots, f_m)^T$$

$$\boldsymbol{\eta} = (\eta_1, \eta_2, \dots, \eta_m)^T$$

T indica el operador transpuesto

En este caso la notación de la fórmula dada en (2.2.2) cambia únicamente a notación vectorial en la f y en la y , quedando como

$$\sum_{j=0}^k \alpha_j \mathbf{y}_{n+j} = h \sum_{j=0}^k \beta_j \mathbf{f}_{n+j} \quad (2.2.5)$$

Una vez más para la elección del paso de integración se requiere del polinomio de estabilidad dado en (2.2.3) pero ahora con la siguiente modificación:

Para $i = 1, \dots, m$ (m el número de ecuaciones diferenciales del sistema) se tienen que verificar las raíces del polinomio de estabilidad:

$$\Pi(r, \bar{h} = h\lambda_i) = \sum_{j=0}^k (\alpha_j - h\lambda_i \beta_j) r^j = \sum_{j=0}^k \alpha_j r^j - h\lambda_i \sum_{j=0}^k \beta_j r^j = \rho(r) - \bar{h} \sigma(r) = 0 \quad (2.2.6)$$

donde

λ_i $i = 1, \dots, m$, es el i -ésimo valor característico de la matriz jacobiana $\frac{\partial f}{\partial y}$ (la cual se está suponiendo constante),
 $\bar{h} = h\lambda_i$

De esta forma si se quiere garantizar un criterio en $\bar{h} = h\lambda$, este debe satisfacerse cuando λ es cualquier valor característico de la matriz jacobiana $\frac{\partial f}{\partial y}$.

Los valores característicos de la matriz jacobiana no son necesariamente números reales. Así en el caso de sistemas, el parámetro $\bar{h} = h\lambda$ que aparece en el polinomio de estabilidad asociado a un método dado, puede ser un número complejo. La definición de estabilidad absoluta queda ahora como:

Definición 2.2.2 Un método multipaso lineal se dice que es absolutamente estable en una región R del plano complejo si, para toda $\bar{h} \in R$, todas las raíces del polinomio de estabilidad $\pi(r, \bar{h})$ asociado con el método satisfacen $|r_s| < 1$ $s=1, 2, \dots, k$.

Así para la elección del paso de integración están involucrados los valores característicos del sistema linealizado del sistema de ecuaciones diferenciales que se va a resolver. Existen sistemas de ecuaciones diferenciales ordinarias conocidos como *stiff* [Lambert] cuyos valores característicos presentan un problema en la elección del paso de integración debido a que se requieren valores muy pequeños de h en la mayoría de los métodos para poder integrar la solución dentro de la zona de estabilidad del método.

II.2.1 Las fórmulas BDF

Las fórmulas BDF (*Backward Differential Formulas*) son métodos multipasos lineales implícitos y fueron creadas para poder resolver sistemas *stiff* éstas están dadas como [Lambert]:

$$h\beta_k y'_{n+k} = \sum_{j=0}^k \alpha_j y_{n+j} \quad (2.2.1.1)$$

Donde:

h es el paso de integración que se avanza la solución

y_{n+k} es el valor de la solución que estamos buscando

y'_{n+k} es la derivada evaluada en el punto y_{n+k}

y_{n+j} con $j = 0, \dots, k-1$ son puntos conocidos de la solución anteriores a y_{n+k}

k es el orden de la fórmula

Los coeficientes β_k y α_j con $j = 0, 1, \dots, k$ asociados a cada una de las fórmulas BDF se presentan en la siguiente tabla [Lambert]:

Orden de la fórmula BDF	Coeficientes asociados							
	β_k	α_6	α_5	α_4	α_3	α_2	α_1	α_0
1	1						1	-1
2	2/3					1	-4/3	1/3
3	6/11				1	-18/11	9/11	-2/11
4	12/25			1	-48/25	36/25	-16/25	3/25
5	60/137		1	-300/137	300/137	-200/137	75/137	-12/137
6	60/147	1	-360/147	450/147	-400/147	225/147	-72/147	10/147

Por ejemplo para el caso de la fórmula BDF de orden 1 ($k=1$) se tiene la siguiente fórmula:

$$hy'_{n+1} = -y_n + y_{n+1} \quad (2.2.1.2)$$

la cual puede escribirse como:

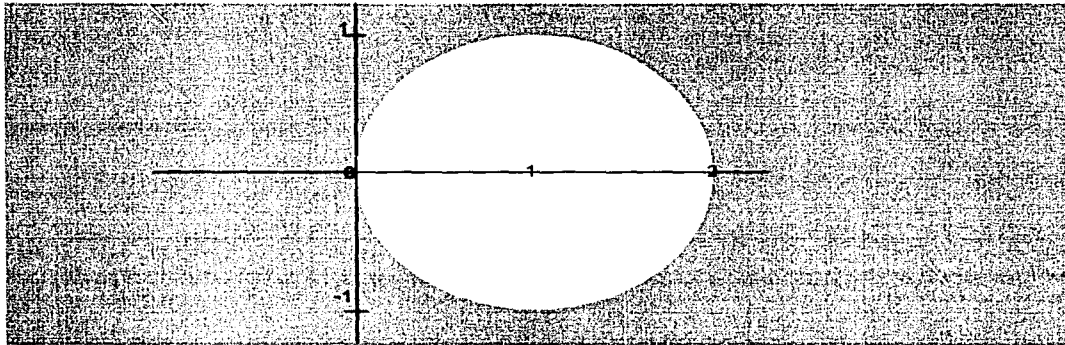
$$y'_{n+1} = \frac{y_{n+1} - y_n}{h} \quad (2.2.1.3)$$

y corresponde al método de Euler hacia atrás.

Esta fórmula es la seleccionada para el desarrollo del presente trabajo, ya que los requerimientos de simulación dinámica de procesos en tiempo real, son mínimos, debido a que los modelos incluyen múltiples aproximaciones y simplificaciones, además algunos

modelos presentan discontinuidades en las derivadas ocasionada por las operaciones de control, lo que justifica el uso de métodos de bajo orden [Rodríguez_1].

La zona de estabilidad absoluta [Lambert] para la fórmula BDF de orden 1, es el complemento del interior y frontera del círculo de radio 1 y centro en la pareja ordenada (1,0) del plano complejo [Gear_1], la cual se muestra en la gráfica 2.2.1.



Gráfica 2.2.1 Zona de Estabilidad Absoluta de la Fórmula BDF de Orden 1 (zona sombreada).

Las zonas de estabilidad para las fórmulas BDF con $k= 2, \dots, 6$ se puede encontrar en [Gear_1], la región de estabilidad para las BDF de orden superior a 6 se hace cada vez más pequeña y por eso las fórmulas BDF con orden superior a 6 no son utilizadas.

Gear [Gear] fue el pionero en utilizar éstas fórmulas ($k=1, \dots, 6$) para la solución de SEAD's en forma simultánea.

II.3 Aplicación de la Fórmula BDF de Orden 1 para la Solución de SEAD's en Forma Simultánea

Para obtener la solución del SEAD (1.5.3), Gear [Gear_2] propone los siguientes puntos:

- I. Considerar como variables diferenciales a las variables algebraicas (z) donde el valor del vector de las derivadas correspondientes a estas variables es igual al vector $\mathbf{0} \in \mathbb{R}^n$. Se denomina como \mathbf{x} el nuevo conjunto de variables diferenciales, que considera tanto a las variables algebraicas como diferenciales del sistema (1.5.3), escribiéndose éste como:

$$\mathbf{F}(\mathbf{x}', \mathbf{x}, t) = \mathbf{0} \quad (2.3.1)$$

Donde en notación matricial tenemos:

$$\mathbf{x} = (y_1, y_2, \dots, y_m, z_1, z_2, \dots, z_n)^T$$

Las y_i $1 \leq i \leq m$ son las componentes del vector original de variables diferenciales dado en (1.5.3) y las z_i con $1 \leq i \leq n$ son las componentes del vector de variables algebraicas dado en (1.5.3).

$$\mathbf{x}' = (y'_1, y'_2, \dots, y'_m, 0_1, 0_2, \dots, 0_n)^T$$

Las y'_i con $1 \leq i \leq m$ son las componentes del vector original de derivadas dado en (1.5.3).

Los 0_i $1 \leq i \leq n$ son los ceros correspondientes a las derivadas de las variables algebraicas.

t

La variable independiente tiempo, $t \in [t_0, t_N]$.

$$\mathbf{F} = (\mathbf{y}' - \mathbf{f}(\mathbf{y}, \mathbf{z}, t), \mathbf{g}(\mathbf{y}, \mathbf{z}, t))^T$$

La función \mathbf{F} está definida en

$$\mathbf{F} : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{m+n}$$

Además de lo anterior se tienen valores iniciales al tiempo t_0 de \mathbf{x}_0 y \mathbf{x}'_0 tales que:

$$\mathbf{F}(\mathbf{x}'_0, \mathbf{x}_0, t_0) = \mathbf{0}$$

- II. Una vez que el SEAD ha sido escrito como (2.3.1), hacer la aproximación de \mathbf{x}' por medio de una fórmula BDF cuyo orden puede ir de 1 hasta 6. Haciendo manipulaciones algebraicas en (2.2.1), la aproximación de la derivada se obtiene como:

$$\mathbf{x}' \approx \frac{\alpha \mathbf{x} - \Psi}{h} \quad (2.3.2)$$

donde \mathbf{x} es la solución buscada, h es el paso de integración con el que se avanza la solución y Ψ es un vector conocido, formado por los valores anteriores de la solución \mathbf{x} .

III. La aproximación (2.3.2) se sustituye en (2.3.1), quedando entonces el SEAD como un sistema de ecuaciones algebraicas que debe resolverse para \mathbf{x} :

$$\mathbf{G}(\mathbf{x}) = \mathbf{F}\left(\frac{\alpha\mathbf{x} - \Psi}{h}, \mathbf{x}, t\right) = \mathbf{0} \quad (2.3.3)$$

utilizando para esto un método iterativo como el método de Newton.

Considerando estas 3 propuestas de Gear, se tiene que:

Para obtener la solución del SEAD (1.5.3), en el intervalo cerrado de tiempo $[T_0, T_F]$, se realizan los siguientes pasos:

- 1.- El intervalo $[T_0, T_F]$ se particiona uniformemente en un conjunto discreto de puntos: $t_0, t_1, \dots, t_n, t_{n+1}, \dots, t_N$ con: $T_0 = t_0$, $T_F = t_N$, $t_{n+1} - t_n = h$, h constante para $n = 0, \dots, N-1$.
- 2.- Se sigue la propuesta I de Gear, y se define un solo vector de variables diferenciales con las variables algebraicas (\mathbf{z}) y las diferenciales (\mathbf{y}), y el SEAD (1.5.3) se lleva así al SEAD (2.3.1).
- 3.- Para avanzar el valor de la solución de (2.3.1) del tiempo t_n al tiempo t_{n+1} , suponiendo que se conoce el valor de la solución al tiempo t_n (denotada como \mathbf{x}_n), se hace una aproximación de la derivada al tiempo t_{n+1} , por medio de la fórmula BDF de orden 1:

$$\mathbf{x}'_{n+1} = \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{h} \quad (2.3.4)$$

Esta aproximación se sustituye en el sistema (2.3.1) quedando como:

$$\mathbf{G}(\mathbf{x}_{n+1}) = \mathbf{F}\left(\frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}, t_{n+1}\right) = \mathbf{0} \quad (2.3.5)$$

con esto, el sistema (2.3.1) se convierte en un sistema de ecuaciones algebraicas para \mathbf{x}_{n+1} , debido a que \mathbf{x}_n es el valor conocido de la solución obtenida en el paso anterior y t_{n+1} es el tiempo conocido al que se avanza la solución. El sistema de ecuaciones (2.3.5) se resuelve por el método de Newton para encontrar \mathbf{x}_{n+1} .

- 4.- Para utilizar el método de Newton, se necesita la obtención de la matriz jacobiana del sistema:

$$\mathbf{F}\left(\frac{\mathbf{x}-\Psi}{h}, \mathbf{x}, t\right) = \mathbf{0} \quad (2.3.6)$$

respecto al vector \mathbf{x} , esta matriz será denotada como \mathbf{JF} .

Para la obtención de esta matriz, obsérvese que el sistema (2.3.6) puede verse como:

$$\mathbf{F}(\mathbf{v}(\mathbf{x}), \mathbf{x}, t) = \mathbf{0}$$

donde

$$\mathbf{v}(\mathbf{x}) = \frac{\mathbf{x} - \Psi}{h}$$

entonces tenemos que

$$\mathbf{JF} = \frac{\partial \mathbf{F}}{\partial \mathbf{v}} \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}$$

donde:

$$\begin{aligned} \frac{\partial \mathbf{F}}{\partial \mathbf{v}} &= \frac{\partial \mathbf{F}}{\partial \mathbf{x}'} \\ \frac{\partial \mathbf{v}}{\partial \mathbf{x}} &= \frac{1}{h} \mathbf{I} \end{aligned}$$

\mathbf{I} es la matriz identidad de $(m+n) \times (m+n)$

Por lo tanto el Jacobiano se obtiene como:

$$\mathbf{JF} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}'} \frac{1}{h} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \quad (2.3.7)$$

- 5.- Utilizando el método de Newton para resolver (2.3.5) y obtener \mathbf{x}_{n+1} la iteración $s+1$ es la siguiente [Conte]:

$$\mathbf{x}_{n+1}^{[s+1]} = \mathbf{x}_{n+1}^{[s]} - \mathbf{JF}^{-1} \mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right) \quad (2.3.8)$$

donde [] indica la iteración de que se trata.

- 6.- La forma presentada en (2.3.8) indica que se requiere de la matriz inversa denotada como \mathbf{JF}^{-1} del jacobiano \mathbf{JF} (2.3.7), lo que implica primero calcular la matriz \mathbf{JF} y luego invertirla, sin embargo, la solución puede obtenerse de una forma equivalente utilizando únicamente la matriz \mathbf{JF} y su descomposición LU [Rodríguez_4]. Los pasos

que se deben seguir para obtener esta solución son:

i) Se expresa (2.3.8) como

$$\mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]} = -\mathbf{JF}^{-1}\mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right) \quad (2.3.9)$$

ii) Se multiplica por la matriz jacobiana \mathbf{JF} ambos lados de la ecuación (2.3.9), y se obtiene

$$\mathbf{JF}(\mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]}) = -\mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right) \quad (2.3.10)$$

iii) Hacer

$$\Delta\mathbf{x}_{n+1}^{[s+1]} = \mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]} \quad (2.3.11)$$

y sustituir (2.3.11) en (2.3.10) para obtener:

$$\mathbf{JF}\Delta\mathbf{x}_{n+1}^{[s+1]} = -\mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right) \quad (2.3.12)$$

iv) Sustituir en (2.3.12) el valor de \mathbf{JF} dado en (2.3.7), se obtiene finalmente el esquema iterativo con el que se resuelven simultáneamente las variables de estado del sistema (2.3.1) por medio de la aplicación de la fórmula BDF de orden 1:

$$\left(\frac{\mathbf{1}}{h} \frac{\partial \mathbf{F}}{\partial \mathbf{x}'} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right) \Delta\mathbf{x}_{n+1}^{[s+1]} = -\mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right) \quad (2.3.13)$$

Donde

[] indica iteraciones, $s=0,1,2,3,\dots$

la matriz jacobiana debe ser evaluada en el mismo punto donde se evalúa la

función \mathbf{F} , es decir en el punto $\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right)$.

II.3.1 Predicción del valor de inicio de la iteración

Para iniciar la iteración del método de Newton al tiempo t_{n+1} , es necesario dar el valor inicial de $\mathbf{x}_{n+1}^{[0]}$. Este valor se obtiene siguiendo la idea dada en [Brayton et al], evaluando en t_{n+1} el polinomio que interpola a los dos últimos puntos de la solución obtenidos \mathbf{x}_n y

x_{n-1} , la predicción obtenida de la solución tendrá el mismo orden de exactitud que la fórmula BDF de orden 1.

Utilizando la forma de Lagrange del polinomio de interpolación de orden 1, éste puede escribirse como:

$$P(t) = \gamma_1(t)x_n + \gamma_2(t)x_{n-1} \quad (2.3.14)$$

donde las $\gamma_i(t)$ $i=1,2$ son los polinomios de interpolación fundamentales:

$$\gamma_1(t) = \left(\frac{t - t_{n-1}}{t_n - t_{n-1}} \right) \quad (2.3.15)$$

$$\gamma_2(t) = \left(\frac{t - t_n}{t_{n-1} - t_n} \right) \quad (2.3.16)$$

Ya que el intervalo de tiempo se divide uniformemente, en puntos discretos $t_0, t_1, \dots, t_{n-1}, t_n, t_{n+1}, \dots, t_N$, donde $t_n - t_{n-1} = h$ para toda $n = 1, \dots, N$,

entonces

$$t_n - t_{n-1} = h \quad (2.3.17)$$

y también se tiene que

$$t_{n-1} - t_n = -h \quad (2.3.18)$$

sustituyendo (2.3.17) y (2.3.18) en (2.3.15) y (2.3.16) respectivamente y evaluando en $t = t_{n+1}$ se obtiene

$$\gamma_1(t_{n+1}) = \frac{t_{n+1} - t_{n-1}}{h} = \frac{2h}{h} = 2 \quad (2.3.19)$$

$$\gamma_2(t_{n+1}) = -\frac{t_{n+1} - t_n}{h} = -\frac{h}{h} = -1 \quad (2.3.20)$$

Así, se llega a que el valor inicial para resolver (2.3.13) obtenido a través de la evaluación en t_{n+1} del polinomio de interpolación de orden 1 que pasa por los dos últimos puntos de la solución obtenida x_n e x_{n-1} es:

$$x_{n+1}^{[0]} = P(t_{n+1}) = \gamma_1(t_{n+1})x_n + \gamma_2(t_{n+1})x_{n-1} \quad (2.3.21)$$

y se obtiene como

$$x_{n+1}^{[0]} = P(t_{n+1}) = 2x_n - x_{n-1} \quad (2.3.22)$$

II.3.2 Algoritmo de la aplicación de la fórmula BDF de orden 1

Para $n=0,1,\dots,N$

1. Dado el valor de la solución x_n al tiempo t_n
2. Predecir $x_{n+1}^{[0]}$ al tiempo t_{n+1} de la siguiente forma:

si $n \geq 1$

$$x_{n+1}^{[0]} = P(t_{n+1}) = 2x_n - x_{n-1}$$

si $n = 0$

$$x_1^{[0]} = x_0$$

3. Predecir $x_{n+1}'^{[0]}$ al tiempo t_{n+1} por medio de

$$x_{n+1}'^{[0]} = \frac{x_{n+1}^{[0]} - x_n}{h}$$

4. Calcular $x_{n+1}^{[s+1]}$ por medio del algoritmo de Newton.

Para $s = 0,1,\dots,m-1$

$$\left(\frac{1}{h} \frac{\partial F}{\partial x'} + \frac{\partial F}{\partial x} \right) \Delta x_{n+1}^{[s+1]} = -F\left(\frac{x_{n+1}^{[s]} - x_n}{h}, x_{n+1}^{[s]}, t_{n+1} \right)$$

$$x_{n+1}^{[s+1]} = x_n^{[s]} + \Delta x_{n+1}^{[s+1]}$$

II.4 Esquema Iterativo Newton para la Solución Simultánea

II.4.1 Aproximación numérica del Jacobiano \mathbf{JF}

La codificación del algoritmo definido en la sección anterior implica la definición de una estrategia que minimice las evaluaciones de la matriz jacobiana dada en (2.3.7):

$$\mathbf{JF} = \frac{1}{h} \frac{\partial \mathbf{F}}{\partial \mathbf{x}'} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}},$$

así como disponer de un método para aproximar numéricamente a ésta. En general, no es práctico en simulación de procesos en tiempo real, solicitar al usuario que la proporcione en forma analítica [Rodríguez_4].

La aproximación numérica por separado de las matrices $\frac{1}{h} \frac{\partial \mathbf{F}}{\partial \mathbf{x}'}$ y $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ para posteriormente sumarlas y así obtener \mathbf{JF} , es una estrategia cuyo costo computacional es alto. Sin embargo, es posible aproximar los elementos de la matriz jacobiana \mathbf{JF} de una forma más eficiente, ya que los elementos de las matrices $\frac{1}{h} \frac{\partial \mathbf{F}}{\partial \mathbf{x}'}$ y $\frac{\partial \mathbf{F}}{\partial \mathbf{x}}$, se pueden calcular una vez que se tienen los valores predichos $\mathbf{x}_{n+1}^{[0]}, \mathbf{x}_{n+1}^{[0]}$ por medio de las siguientes ecuaciones de diferencias [Rodríguez_4]:

$$\frac{1}{h} \frac{\partial F_i}{\partial y_j} \approx \frac{1}{\delta} \left[F_i(\mathbf{x}_{n+1}^{[0]} + \mathbf{e}_j \frac{\delta}{h}, \mathbf{x}_{n+1}^{[0]}, t_{n+1}) - F_i(\mathbf{x}_{n+1}^{[0]}, \mathbf{x}_{n+1}^{[0]}, t_{n+1}) \right] \quad (2.4.1)$$

$$\frac{\partial F_i}{\partial x_j} \approx \frac{1}{\delta} \left[F_i(\mathbf{x}_{n+1}^{[0]} + \mathbf{e}_j \frac{\delta}{h}, \mathbf{x}_{n+1}^{[0]} + \mathbf{e}_j \delta, t_{n+1}) - F_i(\mathbf{x}_{n+1}^{[0]} + \mathbf{e}_j \frac{\delta}{h}, \mathbf{x}_{n+1}^{[0]}, t_{n+1}) \right] \quad (2.4.2)$$

donde \mathbf{e}_j es el vector unitario con la unidad en la j -ésima posición.

Sumando las ecuaciones de diferencias (2.4.1) y (2.4.2), se obtiene que la componente ij -ésima de la matriz jacobiana (2.3.7) se aproxima como [Rodríguez_4]:

$$\frac{1}{h} \frac{\partial F_i}{\partial x_j'} + \frac{\partial F_i}{\partial x_j} \approx \frac{1}{\delta} \left[F_i(\mathbf{x}_{n+1}^{[0]} + \mathbf{e}_j \frac{\delta}{h}, \mathbf{x}_{n+1}^{[0]} + \mathbf{e}_j \delta, t_{n+1}) - F_i(\mathbf{x}_{n+1}^{[0]}, \mathbf{x}_{n+1}^{[0]}, t_{n+1}) \right] \quad (2.4.3)$$

El incremento δ se selecciona de forma tal que minimice la contribución de los errores de redondeo y truncamiento asociados a la computadora usada y a la ecuación de diferencias utilizada para aproximar las derivadas parciales [Rodríguez_2]. En la presente codificación se seleccionó como [Rodríguez_4]:

$$\delta = \sqrt{\frac{\epsilon |x_j|}{5}} \quad (2.4.4)$$

donde ϵ es el épsilon de la computadora usada, esto es, el menor número $\epsilon > 0$ tal que $1 + \epsilon > 1$ en la aritmética de punto flotante de la computadora.

II.4.2 Solución del sistema lineal $\mathbf{JF}\Delta\mathbf{x} = -\mathbf{F}$

La solución del sistema de ecuaciones lineales dado en (2.3.13) como:

$$\mathbf{JF}\Delta\mathbf{x}_{n+1}^{[s+1]} = -\mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right)$$

presente en la parte iterativa del algoritmo de Newton, se realiza por medio de la factorización LU de la matriz jacobiana \mathbf{JF} . Esta factorización es realizada por medio de los programas DECOMP y SOLVE que se describen en [Forsythe].

II.4.3 Criterio de convergencia en las iteraciones del método de Newton

Para analizar la convergencia de las iteraciones del método de Newton, dadas en (2.3.13), es conveniente analizarlas a través de su forma equivalente dada en (2.3.8), las cuales pueden ser escritas como

$$\mathbf{x}_{n+1}^{[s+1]} = \mathbf{G}(\mathbf{x}_{n+1}^{[s]}) \quad (2.4.5)$$

donde

$$\mathbf{G}(\mathbf{x}_{n+1}^{[s]}) = \mathbf{x}_{n+1}^{[s]} - \mathbf{JF}^{-1}\mathbf{F}\left(\frac{\mathbf{x}_{n+1}^{[s]} - \mathbf{x}_n}{h}, \mathbf{x}_{n+1}^{[s]}, t_{n+1}\right)$$

la convergencia de estas iteraciones se basa en el Principio de Contracción y los teoremas de Punto Fijo [Conte]. A continuación se presenta el Principio de Contracción [Rudin].

Definición 2.4.3.1 Sea \mathfrak{R}^n con la norma Euclidiana $\| \cdot \|$. Si φ mapea \mathfrak{R}^n en \mathfrak{R}^n y existe un número r , con $0 < r < 1$ tal que

$$\|\varphi(y) - \varphi(x)\| \leq r\|y-x\|$$

para todo $x, y \in \mathfrak{R}^n$, entonces se dice que φ es una contracción de \mathfrak{R}^n en \mathfrak{R}^n .

De la definición 2.4.3.1 se obtiene el Teorema de Punto Fijo [Rudin] que se enuncia a continuación.

Teorema 2.4.3.1 Si φ es una contracción de \mathfrak{R}^n en \mathfrak{R}^n , entonces existe un y sólo un vector $x \in \mathfrak{R}^n$ tal que $x = \varphi(x)$.

Demostración:

Tómese arbitrariamente $x_0 \in \mathfrak{R}^n$, y defínase la sucesión de puntos $\{x_n\}$ de la siguiente forma recurrente

$$x_{n+1} = \varphi(x_n) \quad \text{con } n=0,1,2,\dots \quad (2.4.6.1)$$

Por ser φ una contracción, existe un número r , con $0 < r < 1$ tal que si $n \geq 1$

$$\|\varphi(x_n) - \varphi(x_{n-1})\| \leq r\|x_n - x_{n-1}\| \quad (2.4.6.2)$$

la cual es equivalente a

$$\|x_{n+1} - x_n\| \leq r\|x_n - x_{n-1}\| \quad (2.4.6.3)$$

por inducción se demuestra que

$$\|x_{n+1} - x_n\| \leq r^n \|x_1 - x_0\| \quad (n=0,1,2,\dots) \quad (2.4.6.4)$$

si $n < m$, tenemos por la desigualdad del triángulo

$$\|x_n - x_m\| \leq \sum_{i=n+1}^m \|x_i - x_{i-1}\| \quad (2.4.6.5)$$

acotando en (2.4.6.5) las componentes de la suma por medio de la expresión dada en (2.4.6.4) tenemos

$$\|x_n - x_m\| \leq (r^n + r^{n+1} + \dots + r^{m-1}) \|x_1 - x_0\| \quad (2.4.6.6)$$

factorizando r^n de la suma $r^n + r^{n+1} + \dots + r^{m-1}$ la expresión (2.4.6.6) se puede escribir como

$$\|x_n - x_m\| \leq r^n (1 + r + \dots + r^{m-1-n}) \|x_1 - x_0\| \quad (2.4.6.7)$$

el término $(1 + r + \dots + r^{m-1-n})$ de la expresión (2.4.6.7) se puede acotar superiormente por medio de la progresión geométrica [Courant] con razón r con $0 < r < 1$, $\sum_{i=0}^{\infty} r^i$, la cual tiene como

límite el valor de $\frac{1}{1-r}$ [Courant], entonces se tiene que

$$\|x_n - x_m\| \leq \frac{r^n}{1-r} \|x_1 - x_0\| \quad (2.4.6.8)$$

dado que $0 < r < 1$ el término r^n de la expresión (2.4.6.8) se puede hacer tan chico como se desee eligiendo n lo suficientemente grande. Con lo anterior se concluye que $\{x_n\}$ es una sucesión de Cauchy [Rudin]. Por ser ésta sucesión de Cauchy y por estar contenida en \mathfrak{R}^n espacio métrico completo, esta sucesión converge [Rudin], es decir $\lim_{n \rightarrow \infty} x_n = x$ para algún $x \in \mathfrak{R}^n$.

Debido a que φ es una contracción, φ es continua sobre \mathfrak{R}^n , por consiguiente

$$\varphi(x) = \lim_{n \rightarrow \infty} \varphi(x_n) = \lim_{n \rightarrow \infty} x_{n+1} = x \quad (2.4.6.9)$$

Para demostrar la unicidad de x , supóngase que existe otro vector $y \in \mathfrak{R}^n$ con $y \neq x$ y $\varphi(y) = y$, entonces de la definición (2.4.3.1) se tiene que

$$\|\varphi(y) - \varphi(x)\| \leq r \|y - x\| \quad \text{con } 0 < r < 1 \quad (2.4.6.10)$$

ahora por ser y y x puntos fijos de φ , la expresión (2.4.6.10) queda como

$$\|y - x\| \leq r \|y - x\| \quad \text{con } 0 < r < 1 \quad (2.4.6.11)$$

lo cual sólo puede ocurrir si $y = x$, y por lo tanto x es el único punto fijo de φ y esto concluye la demostración. \square

Del teorema de Punto Fijo, se obtiene otro teorema que se presenta a continuación, el cual proporciona la fórmula citada por Shampine en [Shampine_2] para verificar la convergencia en las iteraciones del método de Newton.

Teorema 2.4.3.2 Si φ es una contracción de \mathfrak{R}^n en \mathfrak{R}^n , entonces para todo $y \in \mathfrak{R}^n$

$$\|x^* - \varphi(x)\| \leq \frac{r}{1-r} \|\varphi(x) - x\|$$

donde $0 < r < 1$ y x^* es el punto fijo de φ .

Demostración:

Sea φ una contracción de \mathfrak{R}^n en \mathfrak{R}^n , de la definición (2.4.3.1), existe r , con $0 < r < 1$ tal que

$$\|\varphi(y) - \varphi(x)\| \leq r \|y - x\| \quad (2.4.7.1)$$

para todo $\mathbf{x}, \mathbf{y} \in \mathfrak{R}^n$. Por el teorema (2.4.3.1) tenemos que φ tiene un punto fijo al que llamaremos \mathbf{x}^* , remplazando en (2.4.7.1) el punto \mathbf{y} por el punto fijo \mathbf{x}^* tenemos

$$\|\varphi(\mathbf{x}^*) - \varphi(\mathbf{x})\| \leq r \|\mathbf{x}^* - \mathbf{x}\| \quad (2.4.7.2)$$

por el teorema (2.4.3.1) sabemos que $\varphi(\mathbf{x}^*) = \mathbf{x}^*$, entonces sustituyendo en (2.4.7.2) el valor de $\varphi(\mathbf{x}^*)$ obtenemos

$$\|\mathbf{x}^* - \varphi(\mathbf{x})\| \leq r \|\mathbf{x}^* - \mathbf{x}\| \quad (2.4.7.3)$$

ahora bien, en (2.4.7.3) el término $r \|\mathbf{x}^* - \mathbf{x}\| = r \|\mathbf{x}^* - \mathbf{x} + \varphi(\mathbf{x}) - \varphi(\mathbf{x})\|$

aplicando a éste término la desigualdad del triángulo, tenemos que

$$r \|\mathbf{x}^* - \mathbf{x}\| \leq r \|\mathbf{x}^* - \varphi(\mathbf{x})\| + r \|\varphi(\mathbf{x}) - \mathbf{x}\| \quad (2.4.7.4)$$

sustituyendo en (2.4.7.3) la cota encontrada para $r \|\mathbf{x}^* - \mathbf{x}\|$ en (2.4.7.4) tenemos que

$$\|\mathbf{x}^* - \varphi(\mathbf{x})\| \leq r \|\mathbf{x}^* - \mathbf{x}\| \leq r \|\mathbf{x}^* - \varphi(\mathbf{x})\| + r \|\varphi(\mathbf{x}) - \mathbf{x}\| \quad (2.4.7.5)$$

o bien que

$$\|\mathbf{x}^* - \varphi(\mathbf{x})\| \leq r \|\mathbf{x}^* - \varphi(\mathbf{x})\| + r \|\varphi(\mathbf{x}) - \mathbf{x}\| \quad (2.4.7.6)$$

y así se concluye que

$$\|\mathbf{x}^* - \varphi(\mathbf{x})\| \leq \frac{r}{1-r} \|\varphi(\mathbf{x}) - \mathbf{x}\| \quad (2.4.7.7)$$

□

A partir del análisis teórico del Principio de Contracción y de los Teoremas de Punto Fijo mostrado anteriormente, Shampine sugiere en [Shampine_2] que la verificación de la convergencia en las iteraciones del método de Newton se obtenga a través de:

- 1) Verificar que la función \mathbf{G} dada en (2.4.5) sea una contracción, en algún subconjunto \mathbf{A} cerrado de \mathfrak{R}^{n+m} que contenga la solución predicha $\mathbf{x}_{n+1}^{[0]}$, la verificación se obtiene a partir del valor r dado en la definición 2.4.3.1, que en el caso de \mathbf{G} es el valor del $\sup_{\mathbf{x} \in \mathbf{A}} \left\| \frac{\partial \mathbf{G}}{\partial \mathbf{x}} \right\|$ [Rodríguez_4]. Es difícil obtener el valor real de r , sin embargo, a partir de la desigualdad dada en (2.4.6.4), escrita en términos de las iteraciones de Newton que se están realizando, como

$$\|\mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]}\| \leq r^s \|\mathbf{x}_{n+1}^{[1]} - \mathbf{x}_{n+1}^{[0]}\| \quad (s=0,1,2,\dots)$$

se puede obtener una cota inferior ρ_s de r a partir de la segunda iteración por medio de:

$$\rho_s = \left(\frac{\| \mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]} \|}{\| \mathbf{x}_{n+1}^{[1]} - \mathbf{x}_{n+1}^{[0]} \|} \right)^{\frac{1}{s}} \leq r \quad (s=1,2,\dots) \quad (2.4.8)$$

De esta forma si $\rho_s < 1$ ($s=1,2,\dots$), se acepta que la sucesión $\{ \mathbf{x}_{n+1}^{[s]} \}$ es convergente, a partir de la solución predicha $\mathbf{x}_{n+1}^{[0]}$, y en consecuencia si se desea verificar

$$\| \mathbf{x}_{n+1}^* - \mathbf{x}_{n+1}^{[s+1]} \| \leq \varepsilon \quad (2.4.9)$$

para ($s=1,2,\dots$), donde \mathbf{x}_{n+1}^* es el punto fijo buscado de \mathbf{G} y $\varepsilon > 0$, se debe verificar en su lugar:

$$\frac{\rho_s}{1 - \rho_s} \| \mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]} \| \leq \varepsilon \quad (2.4.10)$$

justificada por la desigualdad (2.4.7.9). Un valor de ε es sugerido en [Petzold_2] como 0.33.

En el caso de los SEAD's de procesos dinámicos, la verificación de que $\rho_s < 1$, es relajada, debido a que los SEAD's de los procesos de plantas generadoras de energía eléctrica, presentan discontinuidades, éstas resultan en perturbaciones en la aproximación a la solución \mathbf{x} originando que el error en la primera corrección de \mathbf{x} sea grande y que el valor de ρ_s sea mayor a 1, debido a que en el punto de discontinuidad no se cumple con la existencia de $\sup_{\mathbf{x} \in A} \left\| \frac{\partial \mathbf{G}}{\partial \mathbf{x}} \right\|$. Sin embargo, esta dificultad puede ser en algunos casos controlada en las siguientes correcciones del proceso iterativo, por lo cual la simulación es detenida cuando $\rho_s \geq 2$, enviando un mensaje al usuario en pantalla de "problemas en la convergencia" [Rodríguez_4].

2) Verificar que se cumpla

$$\frac{\| \mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]} \|}{\| \mathbf{x}_{n+1}^{[s]} \|} \leq 10\varepsilon$$

donde ε es el épsilon de la máquina, debido a que si $d_{s+1} = \| \mathbf{x}_{n+1}^{[s+1]} - \mathbf{x}_{n+1}^{[s]} \|$ es relativamente

más pequeño a $\|x_{n+1}^{[s]}\|$ en 10 veces el ϵ de la máquina, d_{s+1} puede ser considerado numéricamente como el vector $\mathbf{0} \in \mathcal{R}^{n+m}$, en esta situación, se acepta el valor de la última iteración como el valor de la solución bajo la suposición que la solución numérica ha sido aproximada a la solución exacta tanto como ha sido posible, dentro de las limitaciones de aproximación de la computadora utilizada. Esta situación se presenta en general, cuando el valor predicho es excelente o cuando la tolerancia en el error de aproximación a la solución exacta es muy exigente [Rodríguez_4, Shampine_2].

Debido a que la solución de los SEAD's se requiere obtener en tiempo real, se desea minimizar el número de evaluaciones de la matriz \mathbf{JF} así como el número de evaluaciones del modelo; esto se logra si la matriz jacobiana se aproxima al tiempo t_0 y posteriormente, se vuelve a aproximar únicamente cuando no hay convergencia.

Por lo que concierne al número de evaluaciones del modelo se permiten un máximo de cuatro iteraciones para alcanzar la convergencia, si ésta no se logra, se verifica si \mathbf{JF} ya ha sido aproximada al tiempo t_n , en caso contrario se aproxima \mathbf{JF} y se intenta de nuevo un máximo de cuatro iteraciones para converger, si a pesar de este reintento no se converge se envía un aviso al usuario de "problemas en la convergencia" y se detiene la simulación [Rodríguez_4].

II.5 Programación del Algoritmo en Fortran

En la presente sección se describen las especificaciones y requerimientos del código EHABDF (Euler Hacia Atrás BDF), así como su diseño y el diagrama de los módulos que lo constituyen. La codificación del algoritmo EHABDF se desarrolló en lenguaje de programación FORTRAN 77 estructurado.

II.5.1 Definición de requerimientos

II.5.1.1 Planteamiento de objetivos

1. El código EHABDF debe aproximar la solución numérica de un conjunto de ecuaciones algebraico-diferenciales (SEAD) del tipo (2.3.1). La solución debe ser avanzada un paso de integración h , a partir del tiempo t , del valor de la solución al tiempo t y del paso de integración h .
2. El número de ecuaciones algebraico-diferenciales que EHABDF debe resolver deberá estar limitado a la capacidad de la memoria y a la rapidez de procesamiento de operaciones numéricas de la máquina que se vaya a utilizar.
3. EHABDF debe modificar el valor de la solución al tiempo $t+h$, pero no modificar el valor del tiempo. Debido a esto, después de cada llamado exitoso a EHABDF el valor del tiempo t deberá ser actualizado por el programa ejecutivo al tiempo $t+h$.
4. En caso de que no se pueda avanzar la solución un paso de integración, EHABDF mandará un mensaje de error a la pantalla indicando por que no se pudo avanzar la solución.
5. Con los puntos 1, 2 y 3 EHABDF puede ser implantado para resolver un sistema de ecuaciones algebraico-diferenciales del tipo (2.3.1) en tiempo real, durante un período de simulación $[T_0, T_F]$ utilizando paso de integración constante h , sin control de error global ni local en la obtención de la solución, a partir del tiempo inicial T_0 y del valor de la solución x_0 al tiempo inicial T_0 .

A continuación se presenta el diagrama conceptual de EHABDF

Sea el SEAD definido en (2.3.1) y sean:

- a) x el valor de la solución de (2.3.1) al tiempo t
- b) h el paso de integración con el cual se avanza la solución a lo largo de la simulación.
- c) t la variable independiente tiempo.

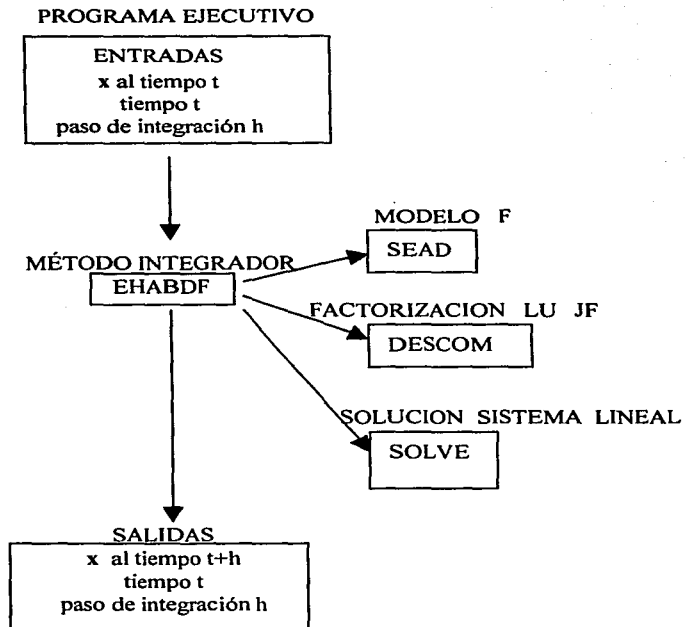


Fig. 2.5.1 Diagrama Conceptual de EHABDF

II.5.1.2 Análisis de objetivos

La solución del SEAD dado en (2.3.1) se obtendrá según lo expuesto en las secciones II.3 y II.4. Sin embargo, la obtención de la solución en tiempo real, dependerá de la rapidez en el procesamiento de operaciones aritméticas de la máquina que se vaya a utilizar y del número de ecuaciones algebraico-diferenciales que se vayan a resolver simultáneamente a través de EHABDF.

Dentro de los principales procedimientos que toman un mayor tiempo en realizarse para la obtención de la solución de (2.3.1) a través del código EHABDF se encuentran:

- La aproximación numérica de la matriz jacobiana (2.3.7), debido a que requiere de la evaluación del modelo para cada una de las perturbaciones realizadas a las variables de estado junto con la perturbación de su correspondiente derivada (2.4.3).

- La descomposición LU de la matriz jacobiana.
- La solución del sistema lineal (2.3.13) utilizando la descomposición LU de la matriz de iteración por medio de la sustitución hacia atrás.
- Las evaluaciones del modelo F dado en (2.3.1) requeridas en (2.3.13).

A continuación se presenta una tabla que muestra el número aproximado de operaciones aritméticas requeridas en los procedimientos mencionados, cuando se resuelve un SEAD de n ecuaciones:

Procedimiento	No. aproximado de operaciones aritméticas requeridas.	Supuestos
Cálculo del Jacobiano.	$n(3+n)$	El SEAD tiene n ecuaciones y solamente se realiza una operación aritmética en cada ecuación del SEAD.
Descomposición LU del Jacobiano.	$\frac{1}{3}n^3$	El jacobiano es de orden n.
Solución del Sistema Lineal por medio de Sustituciones Hacia Atrás.	$\frac{n(n+1)}{2}$	Se cuenta con la descomposición LU del Jacobiano de orden n.
Evaluación del modelo.	n	El modelo (SEAD) cuenta con n ecuaciones y solamente se realiza una operación aritmética en cada ecuación del SEAD.

Analizando la tabla, se puede ver que para resolver un modelo con n ecuaciones algebraico-diferenciales y suponiendo que en cada ecuación se realiza al menos una operación aritmética, entonces se tiene que una cota inferior para el número total de operaciones

requeridas para obtener la solución del SEAD siguiendo el esquema planteado en las secciones II.3 y II.4 es:

$$\frac{1}{3}n^3 + n(n+3) + k\left(\frac{n(n+1)}{2} + n\right) \quad (2.5.1)$$

donde k es el número de veces que se requiere iterar para alcanzar la convergencia de la solución. Así si $k=4$ entonces tenemos:

$$\frac{1}{3}n^3 + n(n+3) + 2n(n+1) + 4n \quad (2.5.2)$$

desarrollando la expresión (2.5.2) tenemos:

$$\frac{1}{3}n^3 + 3n^2 + 9n \quad (2.5.3)$$

En consecuencia, si la máquina tarda t unidades de tiempo en realizar una operación aritmética y se desea obtener la solución del sistema en h segundos tendremos aplicando (2.5.3) que esto se logra sí:

$$t\left(\frac{1}{3}n^3 + 3n^2 + 9n\right) \leq h \quad (2.5.4)$$

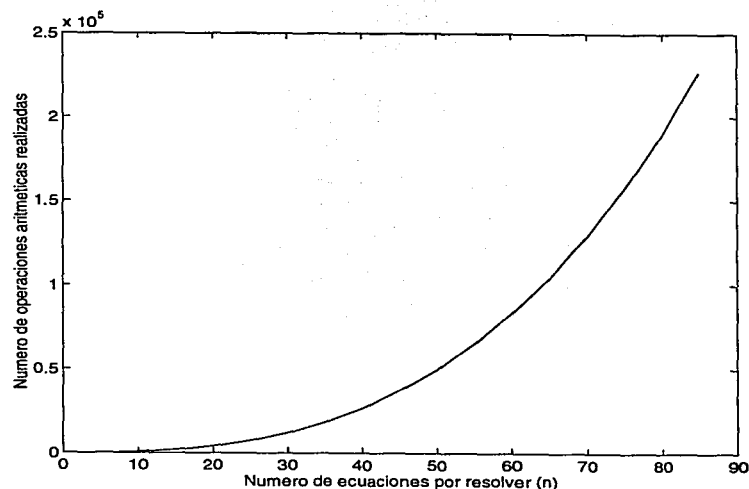
Para la presente tesis será utilizada una estación de trabajo SPARC 10, la cual tiene una rapidez en la realización de una operación aritmética aproximadamente de 0.5 microsegundos [Rodríguez_3], por lo que en nuestro caso $t = 0.000005$ segundos, por otro lado la longitud del intervalo de integración (h) requerido para la obtención de la solución será de 0.125 segundos. Sustituyendo el valor de nuestra t y h en (2.5.4) obtenemos:

$$0.000005\left(\frac{1}{3}n^3 + 3n^2 + 9n\right) \leq 0.125 \quad (2.5.5)$$

ésta expresión puede escribirse como:

$$\frac{1}{3}n^3 + 3n^2 + 9n \leq 250000 \quad (2.5.6)$$

Por lo tanto, el rango del número de ecuaciones que se podrán resolver en tiempo real, bajo estas premisas será de 1 a 87 como lo muestra la gráfica 2.5.1 donde ha sido graficado el lado izquierdo de la desigualdad (2.5.6).



Gráfica 2.5.1 Número de operaciones aritméticas realizadas

II.5.1.3 Especificación de requerimientos

Requerimiento 1.0: **ACTIVACIÓN:** La subrutina EHABDF deberá ser llamada por un programa ejecutivo escrito en lenguaje de programación FORTRAN.

Requerimiento 2.0: **NÚMERO DE ECUACIONES ALGEBRAICO-DIFERENCIALES QUE SE PODRÁN RESOLVER:** El número de ecuaciones que podrán resolverse será desde 1 hasta el número de ecuaciones que la capacidad de memoria y de rapidez en el procesamiento de los cálculos numéricos de la máquina utilizada permita según la expresión dada en (2.5.4).

Requerimiento 3.0: **DIMENSION CORRECTA DE LOS ARREGLOS UTILIZADOS POR EHABDF:** El dimensionamiento de los arreglos utilizados por EHABDF se realizará en el

programa ejecutivo y su dimensión corresponderá al número de ecuaciones algebraico-diferenciales que vayan a resolverse, requerimiento 2.0.

Requerimiento 4.0: INICIALIZACIÓN DE VARIABLES: El programa ejecutivo deberá proporcionar los valores de inicio a las variables que así lo requieran antes de llamar a EHABDF.

Requerimiento 5.0: VALIDACIÓN DE LOS DATOS DE ENTRADA: El programa ejecutivo deberá validar los parámetros de entrada a EHABDF que requieran inicialización, es decir deberá proporcionar:

Paso de integración h constante.

Tiempo t inicial de simulación para cada paso de integración.

Valor numérico x de la solución de (2.3.1) al tiempo t .

Requerimiento 6.0: DEPENDENCIA CON EL SISTEMA OPERATIVO DE LA MÁQUINA: Se debe obtener el valor del épsilon de la máquina (menor número tal que $1 + \epsilon > 1$), ya que este valor debe darse como un parámetro dentro de la subrutina EHABDF.

Requerimiento 7.0: EVALUACIÓN DEL SEAD: Para resolver un SEAD de la forma (2.3.1) a través de EHABDF el usuario deberá escribir en lenguaje FORTRAN una subrutina a la que llamará SEAD, en la cual, defina el sistema de ecuaciones algebraico/diferenciales en forma implícita que quiere resolver. SEAD utilizará como entradas el tiempo t y los vectores x e x' , y dará como salida el vector **RES**, donde $\mathbf{RES} = \mathbf{F}(x', x, t)$, es la cantidad por la cual la función **F** falla en ser cero (**RES** = residuos), para los valores de t , x e x' . La subrutina SEAD debe ser únicamente una forma de evaluación del sistema de ecuaciones, por ningún motivo deberá de modificar el valor de x , x' o t .

Requerimiento 8.0: CÁLCULO DE LA SOLUCIÓN: La subrutina EHABDF se encargará de avanzar el valor de la solución numérica x al tiempo $t+h$.

Requerimiento 8.1: ACTUALIZACION DEL TIEMPO: Inmediatamente después de cada llamado exitoso realizado a la subrutina EHABDF, el programa ejecutivo que coordina el llamado de EHABDF, deberá de actualizar el valor del tiempo a $t+h$, que es el valor del tiempo al cual EHABDF ha aproximado la solución numérica.

Requerimiento 8.2: NO EXISTENCIA DE LA SOLUCIÓN: La subrutina EHABDF no podrá obtener la solución numérica del SEAD (2.3.1) cuando los datos iniciales sean inconsistentes, o la matriz jacobiana sea mal condicionada.

Requerimiento 9.0: PRECISION: La aproximación de la solución x del sistema (2.3.1) obtenida por EHABDF, no requiere mucha exactitud, ya que el objetivo es cumplir con el tiempo real y un cierto grado de aproximación en la obtención de la solución, que será dada por el usuario.

Requerimiento 10.0: RAPIDEZ: La subrutina EHABDF deberá obtener la solución x del sistema (2.3.1) al tiempo $t+h$ en tiempo real, es decir deberá de obtener el valor de x al tiempo $t+h$ en h unidades de tiempo como máximo.

Requerimiento 11.0: IMPRESIÓN DE RESULTADOS: El programa ejecutivo deberá encargarse de imprimir los resultados obtenidos de la simulación que se requieran.

II.5.2 Diseño

II.5.2.1 Arquitectura

El programa EHABDF debe dividirse en las siguientes funciones:

Inicialización de Variables.	<ol style="list-style-type: none">1. Guardar los parámetros y variables de entrada proporcionados por el programa ejecutivo como lo son el valor inicial x_0 de la solución del sistema (2.3.1) y si se desea que en cada paso de integración se calcule ó no la matriz jacobiana.
Predicción de x' al tiempo $t+h$	<ol style="list-style-type: none">1. Predecir el valor de x al tiempo $t+h$, a través del polinomio que interpola a los puntos de la solución del SEAD en t y en $t-h$, y guardar el resultado.2. Predecir el valor de x' al tiempo $t+h$ por medio de la fórmula BDF de orden 1 y guardar la predicción.3. Actualizar $F(x', x, t+h)$ con los valores predichos al tiempo $t+h$ y guardar el valor de los residuos.
Cálculo y Factorización LU de la Matriz Jacobiana a utilizar en el Método de Newton.	<p>Sí se llama a EHABDF por primera vez o si se quiere calcular el jacobiano en cada paso de integración entonces:</p> <p>Aproximar el jacobiano por fórmula de diferencia hacia adelante:</p> <ol style="list-style-type: none">1. Calcular el incremento para la perturbación de x' e x para el cálculo del Jacobiano2. Perturbar x e x'3. Evaluar el SEAD en el punto perturbado4. Estimar la matriz de iteración JF5. Devolver los valores predichos de x e x' al tiempo $t+h$6. Realizar la descomposición LU de la matriz JF por medio de DECOMP

<p>Bloque iterativo para aproximar al vector x. Se define un máximo de cuatro iteraciones.</p>	<p>Bloque iterativo para aproximar al vector x, se define un máximo de cuatro iteraciones para alcanzar la convergencia requerida.</p> <ol style="list-style-type: none"> 1. Iniciar la iteración i. 2. Resolver el sistema de ecuaciones lineal por medio de SOLVE utilizando la descomposición LU de la matriz Jacobiana dada por DECOMP y guardar la solución en un vector. 3. Estimar la norma infinito del vector solución. 4. Actualizar el valor obtenido de x en la iteración i 5. Estimar la norma infinito del vector x. 6. Verificar la convergencia en la iteración i (Ver tabla 2.5.5) <ul style="list-style-type: none"> A partir de la segunda iteración verificar: <ol style="list-style-type: none"> a) Si la constante de Lipchitz es menor que 1. b) Si se está en los límites de precisión de la máquina para aproximar la solución del vector x. c) Si se satisface la tolerancia de error de convergencia en el proceso iterativo para el vector solución. <p>Si no se obtuvo convergencia en la iteración i preparar los valores para una nueva iteración:</p> <ol style="list-style-type: none"> 7. Actualizar las derivadas para la siguiente iteración 8. Actualizar $F(x', x, t)$ para la siguiente iteración 9. Regresar al paso 1 de este bloque con $i = i + 1$
<p>Reporte de no obtención de convergencia después de cuatro iteraciones.</p>	<p>Verificar si ya ha sido calculada la matriz de iteración en el tiempo presente de solución.</p> <p>Si esto es así entonces:</p> <p>Problemas de convergencia, a pesar de haberse recalculado la matriz jacobiana. Regresar al programa ejecutivo el control del flujo informando lo sucedido.</p> <p>Si no ha sido calculada la matriz jacobiana al tiempo actual de solución:</p> <p>Reinicio: Devolver los valores predichos al tiempo $t+h$ y proceder a recalculando la matriz jacobiana inicializando la bandera de cálculo de jacobiano y regresar al bloque de cálculo y descomposición LU de la matriz de iteración.</p>

Tabla 2.5.1 Arquitectura de la subrutina EHABDF.

<p>Verificación de la convergencia en la presente iteración que se realizo del método de Newton</p>	<p>Verificar la convergencia en la iteración i:</p> <p>A partir de la segunda iteración se verifica:</p> <ol style="list-style-type: none"> a) Sí la constante de Lipchitz es menor que 1. b) Sí se está en los límites de precisión de la máquina para aproximar la solución del vector x. c) Sí se satisface la tolerancia de error de convergencia en el proceso iterativo para el vector solución. <ol style="list-style-type: none"> 1. Estimar la constante de Lipchitz 2. Verifica el valor de la constante de Lipchitz 3. Sí la constante de Lipchitz ≥ 2 entonces <ul style="list-style-type: none"> Preguntar si ya ha sido aproximada la matriz jacobiana en el presente tiempo de solución. Sí ya ha sido recalculada la matriz jacobiana entonces: <ul style="list-style-type: none"> No hay convergencia Regresar al programa ejecutivo informando lo sucedido Sí no ha habido recalcule de la matriz jacobiana: <ul style="list-style-type: none"> Calcular una nueva matriz jacobiana e intentar de nuevo las iteraciones. 4. Verificar sí el error relativo cometido en el valor de la solución obtenida en la actual iteración es menor que 10 veces el ϵ de la máquina. <ul style="list-style-type: none"> Sí es así : Límites de precisión de la maquina: no se puede aproximar mas. Regresar al programa ejecutivo con el valor de la solución al tiempo $t+h$ 5. Preguntar sí se satisface el criterio de convergencia <ul style="list-style-type: none"> Sí se satisface el criterio de convergencia, regresar al programa ejecutivo con el nuevo valor de x al tiempo $t+h$.
---	--

Tabla 2.5.2 Arquitectura del proceso de Verificación de Convergencia en la subrutina EHABDF.

II.5.2.2 Diagrama de estructura

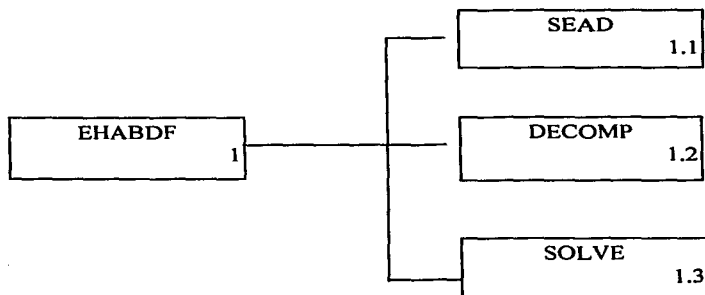


Diagrama de la subrutina EHABDF.

II.5.2.3 Detalle de módulos del sistema.

II.5.2.3.1 Módulo 1: EHABDF

Esta subrutina avanza un paso de integración h , la solución numérica de un sistema de ecuaciones algebraico-diferenciales de la forma (2.3.1), a partir del valor inicial de la solución (x) al tiempo t , de h el paso de integración y del tiempo inicial (t). La solución se aproxima utilizando las ideas planteadas por Gear y Brayton en [Gear_2, Brayton], EHABDF busca aproximar el valor de la solución lo más rápidamente posible, por lo que la matriz jacobiana se mantiene constante el mayor número de pasos de integración posibles, calculándose únicamente al paso de integración inicial y en los pasos de integración en donde no se obtenga convergencia en las iteraciones del método de Newton. Sin embargo, existe la posibilidad de calcular la matriz jacobiana en cada iteración si el usuario así lo requiere.

Si se obtiene convergencia en un número menor o igual al número máximo de iteraciones definidas la variable x saldrá de EHABDF actualizada al tiempo $t+h$, y el tiempo t regresará sin ninguna modificación (el programa principal debe actualizar el tiempo t al valor $t+h$ posteriormente de cada llamado exitoso a EHABDF).

Sí no hay convergencia en el proceso, aún con la actualización de la matriz jacobiana, se mandará un mensaje de error al programa principal y la subrutina EHABDF regresará el control al programa principal el cual deberá detener la simulación.

La subrutina EHABDF debe ser llamada por el programa ejecutivo, en el cual se deben declarar y dimensionar las variables y parámetros que EHABDF necesita como argumentos, así como también, se deben dar valores de inicio a los argumentos que así lo requieran.

A continuación se presentan los argumentos utilizados como entradas por EHABDF

Nombre	Declaración	Descripción
SEAD	EXTERNAL SEAD	Subrutina que contiene el sistema de ecuaciones algebraico-diferenciales (ver sección II.5.2.3.2).
H	REAL*4 H	Parámetro que guarda el paso de integración numérica.
HINV	REAL*4 HINV	Parámetro que almacena el inverso multiplicativo del paso de integración.
T	REAL*4 T	Variable que almacena la variable independiente tiempo.
BANIN	LOGICAL BANIN	Bandera lógica, indica a EHABDF cuando se requiere calcular una matriz jacobiana. Ésta se debe declarar .TRUE. en el programa principal, la primera vez que se llama al método integrador ó cuando el usuario quiere que en el paso de integración que se va a realizar se calcule la matriz jacobiana.
PRIMPAS	LOGICAL PRIMPAS	Bandera lógica, que indica a EHABDF la ocurrencia del primer paso de integración que se va avanzar la solución. Ésta se debe declarar .TRUE. en el programa principal, la primera vez que se llama a EHABDF.
NEDAS	INTEGER*4 NEDAS	Parámetro que indica el número total de ecuaciones algebraico-diferenciales que se van a resolver.
X	REAL*4 X(NEDAS)	Arreglo que contiene a las variables de estado y algebraicas del sead como en (2.3.1).
DX	REAL*4 DX(NEDAS)	Arreglo que contiene las derivadas de las variables de estado y algebraicas como en (2.3.1).
PSI	REAL*4 PSI(NEDAS)	Arreglo auxiliar donde EHABDF guarda el valor de la solución X al tiempo inicial de entrada T.

Nombre	Declaración	Descripción
PSIPA	REAL*4 PSIPA(NEDAS)	Arreglo auxiliar donde EHABDF almacena el valor de la solución X al tiempo T-H (un paso de integración antes al tiempo de entrada T).
XPRED	REAL*4 XPRED(NEDAS)	Arreglo auxiliar en el que EHABDF guarda el valor predicho de la solución X al tiempo T+H.
DXPRED	REAL*4 DXPRED(NEDAS)	Arreglo auxiliar en el que EHABDF guarda el valor predicho de X' al tiempo T+H.
RES	REAL*4 RES(NEDAS)	Arreglo en que se almacenan el valor de los residuos del SEAD al tiempo T+H (RES = F(DX, X, T+H)).
TEMPRES	REAL*4 TEMPRES(NEDAS)	Arreglo auxiliar en que se guarda el valor de los residuos del SEAD una vez que han sido perturbados las variables necesarias para poder calcular la matriz jacobiana al tiempo T+H, si ésta es requerida al tiempo T+H.
MATJAC	REAL*4 MATJAC(NEDASC)	Arreglo auxiliar en el que EHABDF guarda el valor de la matriz jacobiana (2.3.7) para las iteraciones del método de Newton. Nota aclaratoria NEDASC=NEDAS*NEDAS.
DEL	REAL*4 DEL(NEDAS)	Arreglo auxiliar donde se almacena (2.3.11).
IPVT	INTEGER*4 IPVT(NEDAS)	Arreglo auxiliar donde se guardan los índices de los pivotes en la solución del sistema lineal (2.3.13).

II.5.2.3.2 Módulo 1.1:SEAD

Subrutina que contiene las ecuaciones del sistema de ecuaciones algebraico-diferenciales (2.3.1), escrito en el siguiente formato:

$$RES=F(DX, X, T)$$

por lo cual, dados los valores de DX, X y T, evalúa el sistema de ecuaciones algebraico-diferencial guardando el resultado de la evaluación del sead (2.3.1) en el arreglo RES.

Esta subrutina no modifica el valor de DX, X ó T.

A continuación se presentan los argumentos de entrada que utiliza la subrutina SEAD.

Nombre	Declaración	Descripción
T	REAL*4 T	Variable independiente tiempo
NEDAS	INTEGER*4 NEDAS	Parámetro que indica el número total de ecuaciones algebraico-diferenciales que se van a resolver.
X	REAL*4 X(NEDAS)	Arreglo que contiene a las variables de estado y algebraicas del sead como en (2.3.1).
DX	REAL*4 DX(NEDAS)	Arreglo que contiene las derivadas de las variables de estado y algebraicas como en (2.3.1).

Argumentos de salida (variables que son modificadas después de llamar a SEAD).

Nombre	Declaración	Descripción
RES	REAL*4 RES(NEDAS)	Arreglo que guarda la evaluación del SEAD (2.3.1) dados DX, X y T.

II.5.2.3.3 Módulo 1.2:DECOMP

Realiza la factorización LU de la matriz real **JF** (jacobiano), por medio de eliminación Gaussiana. La factorización se escribe en la matriz **JF**. Para mayor detalle consultar [Forsythe].

A continuación se presentan los argumentos de entrada que utiliza la subrutina DECOMP

Nombre	Declaración	Descripción
NDIM	INTEGER*4 NDIM	Número de renglones de la matriz JF.
N	INTEGER*4 N	Número de columnas de la matriz JF.
JF	REAL*4 JF(NDIM*NDIM)	Arreglo que en la entrada almacena los elementos de la matriz del sistema lineal.
IPVT	INTEGER*4 IPVT(NDIM)	Arreglo de trabajo que almacena los índices de los pivotes utilizados en la descomposición LU.

Argumentos de salida (variables que son modificadas después de llamar a DECOMP)

Nombre	Declaración	Descripción
JF	REAL*4 JF(NDIM*NDIM)	Arreglo que a la salida almacena los elementos de la matriz triangular inferior (L) y la matriz triangular superior (U), en que se factorizó la matriz del sistema.
IPVT	INTEGER*4 IPVT(NDIM)	Arreglo de trabajo que almacena los índices de los pivotes utilizados en la descomposición LU.

II.5.2.3.4 Módulo 1.3: SOLVE

Resuelve el sistema de ecuaciones lineales $\mathbf{JF}\Delta\mathbf{x} = \mathbf{RES}$, utilizando la descomposición LU de la matriz JF proporcionada por la subrutina DECOMP la solución del sistema lineal $\Delta\mathbf{x}$ se regresa en el vector **RES**. Para mayor detalle consultar [Forsythe].

A continuación se presentan los argumentos de entrada que utiliza la subrutina SOLVE.

Nombre	Declaración	Descripción
NDIM	INTEGER*4 NDIM	Número de renglones de la matriz JF.
N	INTEGER*4 N	Número de columnas de la matriz JF.
JF	REAL*4 JF(NDIM*NDIM)	Arreglo que contiene la descomposición LU de la matriz JF, proporcionada por la subrutina DECOMP
RES	REAL*4 RES(NDIM)	Arreglo que de entrada contiene el valor del vector RES.
IPVT	INTEGER*4 IPVT(NDIM)	Arreglo proporcionado por DECOMP que contiene el valor de los índices de los pivotes utilizados en la descomposición LU de la matriz JF.

Argumentos de salida (variables que son modificadas después de llamar a SOLVE):

Nombre	Declaración	Descripción
RES	REAL*4 RES(NDIM)	A la salida RES guarda el valor de la solución del sistema lineal.

II.6 Validación Inicial del Algoritmo EHABDF

II.6.1 Modelos elegidos para la validación inicial

Para validar la correcta implantación del código EHABDF se seleccionaron tres modelos:

- 1) Un sistema diferencial lineal $x' = Ax$.
- 2) Un sistema de segundo orden bajo la acción de una función de fuerza externa [Rodríguez_4].
- 3) Las ecuaciones de Euler [Rodríguez_4].

Estos modelos se presentan en la siguiente tabla:

No Mod.	Modelo	Valores Iniciales de las Variables Edo.	Solución Analítica
1	$\begin{aligned} y_1' &= y_2 \\ y_2' &= -2y_1 - 3y_2 \end{aligned}$	$\begin{aligned} y_1(0) &= 0 \\ y_2(0) &= 1 \end{aligned}$	$\begin{aligned} y_1(t) &= e^{-t} - e^{-2t} \\ y_2(t) &= -e^{-t} + 2e^{-2t} \end{aligned}$
2	$x'' + 6x' + 5x = f(t)$ <p>donde</p> $f(t) = \begin{cases} 0, & \text{si } t < 1 \\ 1, & \text{si } 1 \leq t < 2 \\ 0, & \text{si } t \geq 2 \end{cases}$	$x''(0) = x'(0) = x(0) = 0$	$x(t) = y(t) - y(t-1)$ <p>donde</p> $y(t) = \begin{cases} 0, & \text{si } t < 1 \\ \frac{1}{5} - \frac{1}{4}e^{-(t-1)} + \frac{1}{20}e^{-5(t-1)}, & \text{si } t \geq 1 \end{cases}$
3	$\begin{aligned} y_1' &= y_2y_3 \\ y_2' &= -y_1y_3 \\ y_3' &= -0.51y_1y_2 \end{aligned}$	$\begin{aligned} y_1(0) &= 0.0 \\ y_2(0) &= 1.0 \\ y_3(0) &= 1.0 \end{aligned}$	No se dispone de ella

Tabla 2.6.1 Modelos elegidos para la primera validación del código EHABDF

Los motivos para la elección de estos modelos, se presentan en la siguiente tabla:

No Mod.	Motivos de elección
1	<p>Este modelo cuenta con una solución analítica con la cual se puede comparar directamente el valor de la solución numérica obtenida por el código EHABDF. También se conocen los valores característicos asociados a este sistema de ecuaciones diferenciales, los cuales caen dentro de la zona de estabilidad de la fórmula BDF de orden 1, los valores característicos son:</p> $\lambda_1 = -1$ $\lambda_2 = -2$
2	<p>En este modelo se presentan dos discontinuidades de primer orden en la función externa $f(t)$ y se desea analizar el comportamiento del código EHABDF ante tales discontinuidades. También se cuenta con la solución analítica del sistema para su comparación con los resultados obtenidos por EHABDF además se conocen los valores característicos asociados a este sistema de ecuaciones diferenciales, los cuales caen dentro de la zona de estabilidad de la fórmula BDF de orden 1, los valores característicos son:</p> $\lambda_1 = -5$ $\lambda_2 = -1$
3	<p>Los valores característicos asociados a éste sistema (ver gráfica 2.6.6.1) están fuera de la zona de estabilidad absoluta de la fórmula BDF de orden 1 [Rodríguez_4], y se quiere ver el comportamiento del algoritmo bajo esta circunstancia. En este caso no disponemos de una solución analítica, sin embargo se realizó una simulación con el código ODE [Shampine_1] para elaborar las comparaciones.</p>

Tabla 2.6.2 Motivos para la elección de los modelos elegidos para la primera validación del código EHABDF

II.6.2 Pruebas realizadas

Las pruebas realizadas al código EHABDF consistieron en realizar simulaciones de los modelos con diferentes pasos de integración durante un período de 10 s., ya que durante este período de tiempo ocurren los eventos que se desean observar del comportamiento del código respecto a los problemas que plantea cada modelo. Los modelos se resolvieron con el código EHABDF utilizando los siguientes parámetros de simulación:

Modelo No.	Pasos de integración utilizados	Tiempo de Simulación (s)	Error de tolerancia	No. máximo de iteraciones	Cálculo de jacobiano
1	1)0.125 2)0.25 3)0.5 4)1	10	0.33	4	Cuando EHABDF lo requiera.
2	1)0.0125 2)0.125 3)0.25 4)0.5 5)1	10	0.33	4	Cuando EHABDF lo requiera.
3	1)0.125 2)0.25 3)0.5	10	0.33	4	Cuando EHABDF lo requiera.

II.6.3 Análisis de los resultados

El error en la aproximación numérica obtenido con el código EHABDF, se midió como el error absoluto o relativo cometido en la solución obtenida de los diferentes modelos descritos en la sección II.6.1. En los modelos (1) y (2) el error se obtuvo a través de la comparación del valor de la solución analítica en cada paso de integración alcanzado exitosamente con el valor obtenido por el código EHABDF. En estos modelos, los errores se midieron como errores absolutos, debido a que los valores absolutos de la solución analítica en ambos modelos son menores o iguales a 1.

Para el modelo (3), ya que no se cuenta con una solución analítica con la cual comparar los resultados obtenidos, se utilizó el código ODE de Shampine [Shampine_1], el cual proporciona resultados muy precisos, para generar la base de referencia del valor exacto de la solución del modelo, y así comparar los resultados obtenidos con el código EHABDF. Los errores medidos en la solución de este modelo, son errores absolutos debido a que los valores absolutos de las variables reportados por ODE, fueron menores o iguales a 1.

El costo computacional se midió con el número de evaluaciones realizadas al modelo y con el número de matrices de iteración requeridas para la obtención de la solución del modelo.

En las siguientes secciones se presentan la aproximación numérica y el costo computacional realizado por el código EHABDF en la solución de los modelos elegidos.

II.6.4 Resultados Modelo No. 1

II.6.4.1 Aproximación numérica

Los máximos errores absolutos cometidos en las variables de estado y_1 y y_2 del modelo (1) durante las 4 simulaciones realizadas, se presentan en la Tabla 2.6.4.1 dependiendo del paso de integración utilizado:

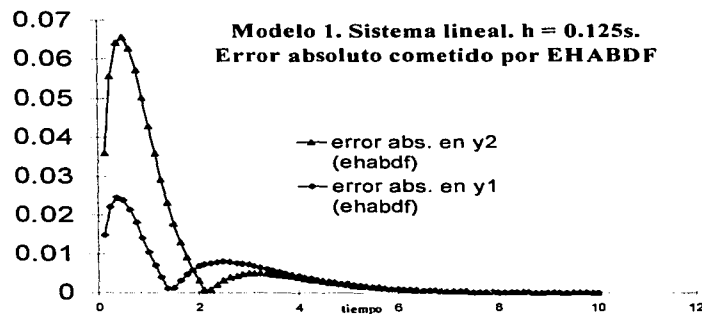
Paso De Integración	Máximo Error Absoluto	Tiempo donde se cometió éste error (s)	Máximo Error Absoluto	Tiempo donde se cometió éste error (s)
	y_1	y_1	y_2	y_2
0.125	0.02459	0.375	0.06568	0.5
0.25	0.04310	0.5	0.11966	0.5
0.5	0.07198	0.5	0.20411	0.5
1	0.06588	1	0.26388	1

Tabla 2.6.4.1 Máximos errores absolutos cometidos en la solución del modelo 1

Como puede verse en la Tabla 2.6.4.1, los máximos errores absolutos cometidos en las variables de estado se encuentra en los primeros pasos de integración. Los errores absolutos cometidos en la variable y_2 fueron en cada simulación los más grandes. Sin embargo, la tendencia de la solución de y_2 se conserva para los pasos de integración de 0.125 s y 0.25 s. El error absoluto máximo crece si se aumenta el paso de integración o disminuye si el paso de integración disminuye, debido al error de discretización del método.

La Gráfica 2.6.4.1 muestra el error cometido en cada paso de integración de 0.125 s. alcanzado exitosamente en la solución del sistema (1), como se puede observar, en ambas variables de estado, el mayor error absoluto cometido, se encuentra alrededor de los dos primeros segundos de simulación, en estos dos segundos el error alcanza su valor máximo a partir del cual empieza a decrecer; posteriormente a estos dos segundos el error vuelve a

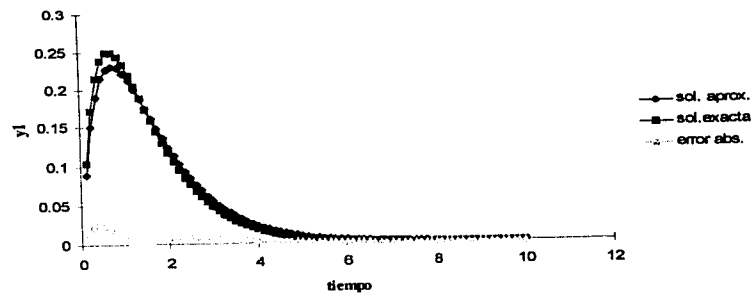
aumentar llegando a otro máximo a partir del cual vuelve a declinar hasta tender a cero conforme transcurre la simulación. Este comportamiento de los errores absolutos mostrados en la gráfica para el paso de integración de 0.125 s, fue similar al resto de los pasos de integración utilizados.



Gráfica 2.6.4.1 Errores absolutos cometidos por EHABDF al solucionar el modelo (1) $h=0.125$ s

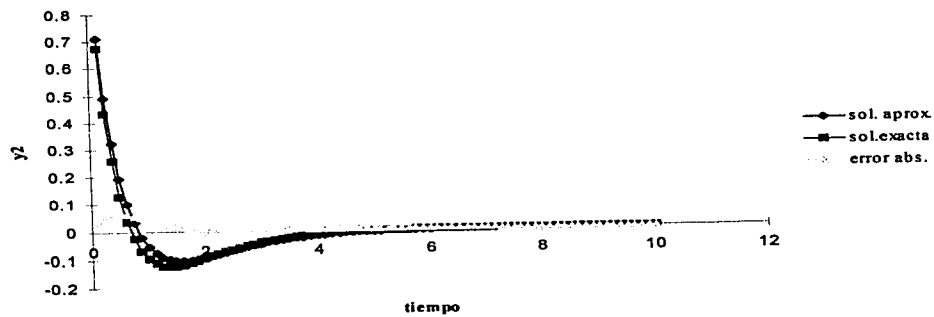
Las gráficas (2.6.4.2) y (2.6.4.3) muestran la solución exacta, la solución obtenida por EHABDF y el error absoluto cometido en la obtención de las variable y_1 y y_2 respectivamente, cuando el modelo se solucionó con un paso de integración de 0.125 s. Las gráficas muestran como la solución obtenida por EHABDF en ambas variables, sigue las tendencias cualitativas de su solución exacta.

Modelo 1. Sistema lineal simulado durante 10 s utilizando paso de integración de 0.125s



Gráfica 2.6.4.2 Solución exacta vs Solución Aproximada por EHABDF en la variable y_1 del modelo 1, $h=0.125$

Modelo 1. Sistema lineal simulado durante 10 s. paso de integración utilizado de 0.125



Gráfica 2.6.4.3 Solución exacta vs Solución Aproximada por EHABDF en la variable y_2 del modelo 1, $h=0.125$

II.6.4.2 Costo computacional

El costo computacional realizado por EHABDF para los diferentes pasos de integración con que se resolvió el modelo se resume en la siguiente tabla:

Paso De Integración	Matrices Jacobianas Calculadas en toda la simulación	Tiempos de cálculo de la Matriz Jacobiana.	No. Evaluaciones del modelo al inicio de la simulación.	No. Evaluaciones del modelo en cada paso de integración posterior al paso inicial de la simulación
0.125	1	0.125	4	2
0.25	1	0.25	4	2
0.5	1	0.5	4	2
1	1	1	4	2

Tabla 2.6.4 Costo computacional cometido en la solución del modelo I

La matriz jacobiana se cálculo solamente una vez en cada una de las simulaciones efectuadas, el cálculo se realizó en el primer paso de integración con el que se avanzó la solución del modelo. La matriz jacobiana del modelo es de hecho constante, debido a que se trata de un sistema lineal con coeficientes constantes.

En relación al número de evaluaciones realizadas al modelo, se obtuvo que en cada una de las simulaciones efectuadas, en el primer paso tomado, el modelo se evaluó 4 veces, debido a que en éste paso es necesario calcular la matriz jacobiana, para ello se necesitan tantas evaluaciones del modelo como el número de ecuaciones a resolver, en este caso son 2 ecuaciones. Una vez calculada la matriz jacobiana, se requiere realizar las iteraciones para obtener la solución del modelo, las cuales, están forzadas por el método de ser al menos 2, por lo que se observa que en el primer paso de integración se alcanzó convergencia en 2 iteraciones, ya que en este paso se realizaron 4 evaluaciones del modelo. Para el resto de los pasos de integración se requirieron 2 iteraciones para alcanzar la convergencia de la solución por lo que el modelo se evaluó dos veces en estos pasos.

II.6.5 Resultados Modelo No. 2

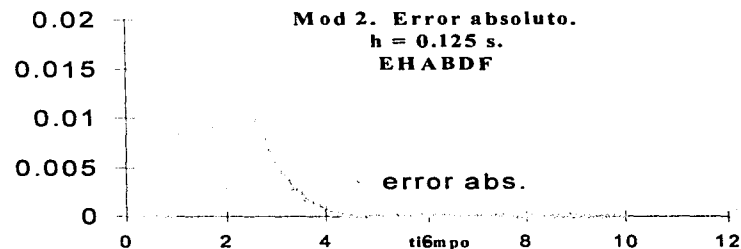
II.6.5.1 Aproximación numérica

El máximo error absoluto cometido en la variable de estado x del modelo (2) durante las 5 simulaciones realizadas, se presenta en la Tabla 2.6.5.1 dependiendo del paso de integración utilizado:

Paso de Integración	Máximo Error Absoluto x	Tiempo donde se cometió este error (s) x
0.0125	0.00232	2.2625
0.125	0.01644	1.25
0.25	0.03025	1.25
0.5	0.04762	1
1	0.08333	1

Tabla 2.6.5.1 Máximos errores absolutos cometidos en la solución del modelo 2

Se observa en la tabla que los mayores errores absolutos se cometen alrededor del tiempo 1 s. de simulación. Recordemos (ver sección II.6.1), que es a partir del tiempo 1 s. donde la función $f(t)$ presenta una discontinuidad al cambiar su valor de 0 a 1, es por ello que EHABDF tiene alrededor de este tiempo los mayores errores absolutos. En la tabla también se muestra que al aumentar el paso de integración el error absoluto también se incrementa. La gráfica 2.6.5.1 muestra el error cometido en cada paso de integración de 0.125 s alcanzado exitosamente por EHABDF.

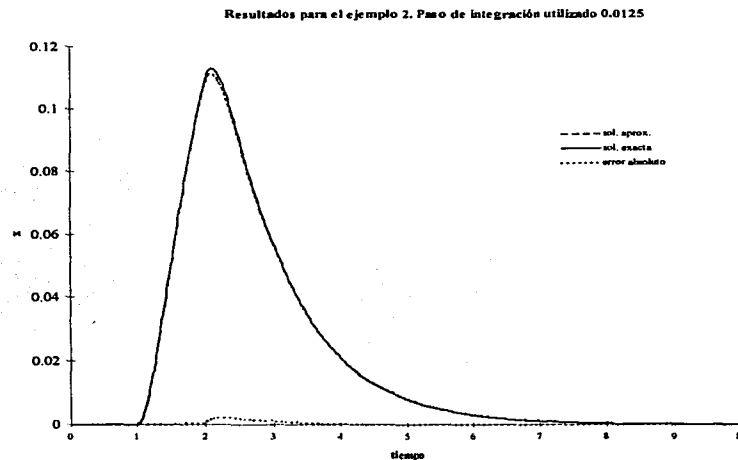


Gráfica 2.6.5.1

En la Gráfica 2.6.5.1 podemos observar como es a partir del primer segundo de simulación tomado y hasta llegar al 4º segundo donde se cometen los mayores errores absolutos. Esto se debe, a las dos discontinuidad de primer orden que presenta la función $f(t)$ en el intervalo

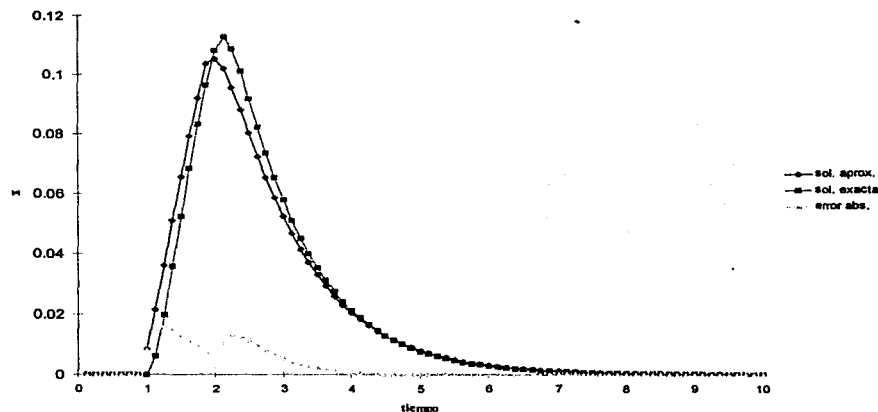
[1,2]. Podemos observar que a partir del 4º segundo de simulación el error se mantiene por debajo de 0.00125, este comportamiento del error fue similar para el resto de los pasos de integración utilizados, solo que el error aumenta conforme aumenta el paso de integración.

Las gráficas (2.6.5.2) y (2.6.5.3) muestran a la solución exacta, a la solución obtenida por EHABDF y al error absoluto cometido en la obtención de la solución del modelo 2, cuando se utilizaron los pasos de integración de 0.0125 y 0.125 segundos respectivamente. Las gráficas muestran como la solución obtenida por EHABDF con paso de integración de 0.0125 s., es muy parecida a la solución exacta, y como la solución obtenida con paso de integración de 0.125 s., sigue las tendencias cualitativas de su solución exacta, también se puede apreciar que el error absoluto cometido es mayor en el intervalo de tiempo [1,4].



Gráfica 2.6.5.2 Solución exacta vs Solución Aproximada por EHABDF en la variable x del modelo 2, $h=0.0125$

Mod 2. Sistema de segundo orden con la introducción de un pulso unitario. Resultados para x . $h = 0.125$ s.



Gráfica 2.6.5.3 Solución exacta vs Solución Aproximada por EHABDF en la variable x del modelo 2, $h = 0.125$

II.6.5.2 Costo computacional

El costo computacional realizado por EHABDF para las 5 simulaciones realizadas se resume en la siguiente tabla:

Paso de Integración	Matrices Jacobianas calculadas en toda la simulación	Tiempos de cálculo de la Matriz Jacobiana.	No. Evaluaciones del modelo al inicio de la simulación.	No. Evaluaciones del modelo en cada paso de integración posterior al paso inicial de la simulación
0.0125	1	0.0125	4	2
0.125	1	0.125	4	2
0.25	1	0.25	4	2
0.5	1	0.5	4	2
1	1	1	4	2

Tabla 2.6.5.2 Costo computacional cometido en la solución del modelo 2

La matriz jacobiana se cálculo una sola vez en cada una de las simulaciones efectuadas, el cálculo se realizó al primer paso de integración con el que se avanzó la solución del

modelo; esta matriz se mantuvo constante a lo largo de la simulación por lo cual no hubo necesidad de aproximarla de nuevo.

En relación al número de evaluaciones del modelo, en cada una de las simulaciones efectuadas, en el primer paso tomado, el modelo se evaluó 4 veces, debido a que en éste paso es necesario calcular la matriz jacobiana, para ello se necesitan tantas evaluaciones del modelo como el número de ecuaciones a resolver, en este caso son 2 ecuaciones ya que el sistema se resolvió transformándolo en un sistema de 2 ecuaciones de primer orden. Una vez calculada la matriz jacobiana, se requieren realizar las iteraciones para obtener la solución del modelo, las cuales, están forzadas por el método de ser al menos 2, por lo que se observa que en éste paso de integración se alcanzó convergencia en 2 iteraciones. Para el resto de los pasos de integración se requirió de 2 evaluaciones del sistema, para alcanzar la convergencia de la solución.

II.6.6 Resultados Modelo No. 3

II.6.6.1 Aproximación numérica

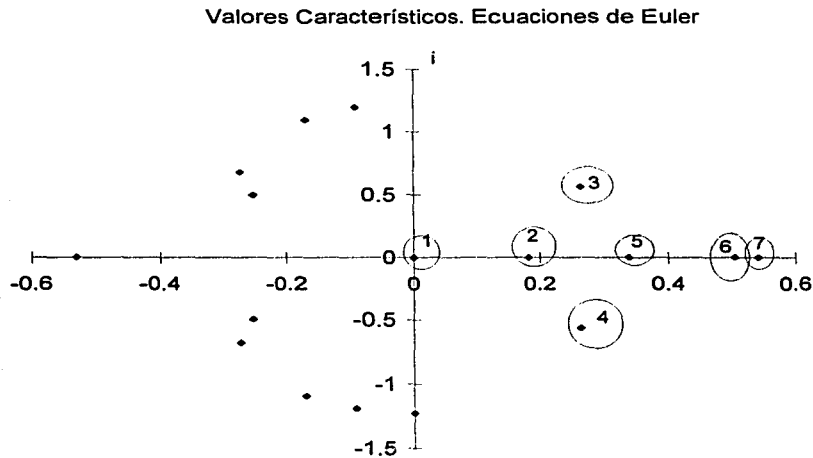
Como se ha citado anteriormente, se utilizó el código ODE como referencia de comparación para los resultados obtenidos por EHABDF de este modelo, debido a que no se cuenta con la solución analítica de este modelo.

A continuación se presenta los parámetros con los cuales cálculo ODE la solución de este modelo.

Tiempo Inicial de Simulación. (segundos)	Tiempo Final de Simulación. (segundos)	Error Absoluto	Error Relativo	Puntos de Salida
0	10	10^{-6}	10^{-6}	a) 0.125 segundos b) 0.25 segundos c) 0.5 segundos

Como se dijo previamente, la elección de este modelo, se debió a que los valores característicos asociados a este sistema, se encuentran fuera de la zona de estabilidad de la fórmula *bdf* de orden 1.

Los valores característicos obtenidos con la solución del código ODE en diferentes puntos de la simulación se presentan en la Gráfica 2.6.6.1, en ella podemos observar como hay valores característicos fuera de la zona de estabilidad de la fórmula BDF de orden 1, dicha zona de inestabilidad es la frontera y el interior del círculo de radio 1 con centro en (1,0) contenido en el plano complejo.



Gráfica 2.6.6.1 Valores característicos asociados a las ecuaciones de Euler.

La presencia de estos valores característicos hace que la solución numérica obtenida no sea muy buena como se puede observar en la tabla 2.6.6.1, en esta se presentan los máximos errores absolutos cometidos en las variables de estado y_1 , y_2 y y_3 del modelo (3) durante las 4 simulaciones realizadas, dependiendo del paso de integración utilizado:

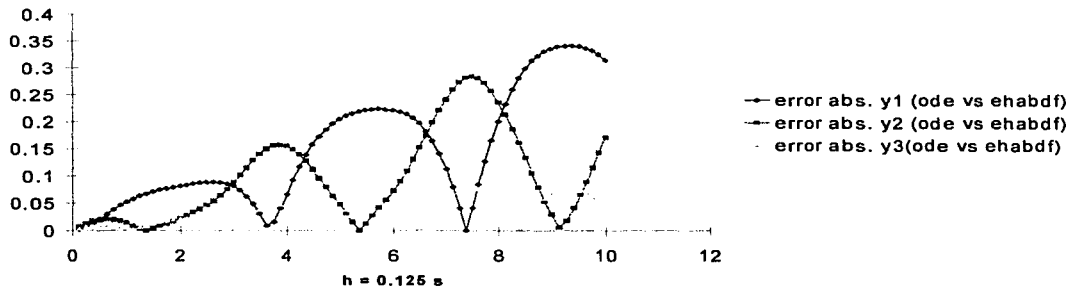
Paso De Integración	Máximo Error Absoluto y_1	Tiempo donde se Cometi6 este error (s) y_1	Máximo Error Absoluto y_2	Tiempo donde se Cometi6 este error (s) y_2	Máximo Error Absoluto y_3	Tiempo donde se cometi6 este error (s) y_3
0.125	0.341374	9.375	0.284739	7.5	0.079027	7.5
0.25	0.56927	9.75	0.489089	7.75	0.120767	7.5
0.5	0.819786	9	0.746509	7.5	0.150947	7.5

Tabla 2.6.6.1 Máximos errores absolutos cometidos en la solución del modelo 3

Se puede observar en la tabla anterior, que el máximo error absoluto cometido en la solución del modelo 3, a través de EHABDF, utilizando paso de integración de 0.125 es de 0.341374, 0.284739 y de 0.079027 para las variables y_1 , y_2 y y_3 respectivamente, estos errores se incrementan al aumentar el tamaño de paso de integración.

La gráfica 2.6.6.2 muestra el error cometido por EHABDF en la solución del modelo (3) en cada paso de integración de 0.125 s alcanzado exitosamente.

Error en la aproximación numérica. Ecs. Euler.

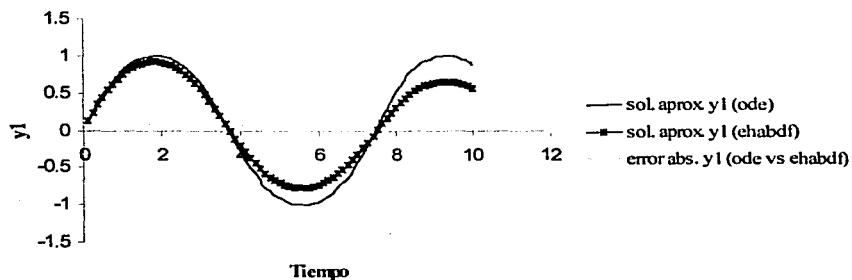


Gráfica 2.6.6.2

Como puede verse, el error comienza a aumentar conforme se avanza la solución en el tiempo de simulación, este patrón de comportamiento se sigue en el resto de los pasos de integración utilizados. Lo cual se esperaba en virtud de que los valores característicos asociados a este modelo como se indicó previamente se encuentran fuera de la zona de estabilidad de la fórmula BDF de orden 1.

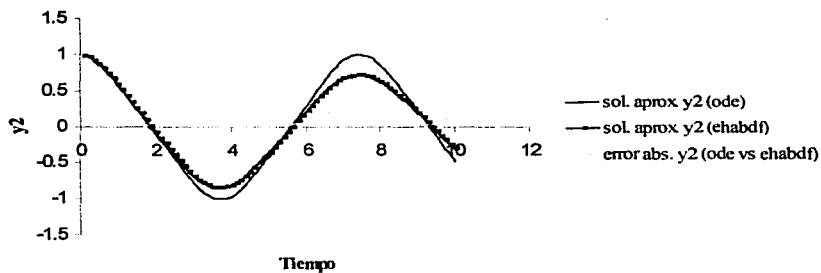
Las gráficas (2.6.6.3), (2.6.6.4) y (2.6.6.5) presentan cada una, la solución obtenida con ODE, la solución obtenida con EHABDF y el error cometido, en la obtención de las variables del modelo no. 3, y_1 , y_2 y y_3 respectivamente, cuando se utilizó el paso de integración de 0.125 segundos. En ellas se aprecia como la solución obtenida por EHABDF se aleja de la solución obtenida por ODE al transcurrir el tiempo de simulación.

Mod. 3 Ecuaciones de Euler, $h=0.125$ s.



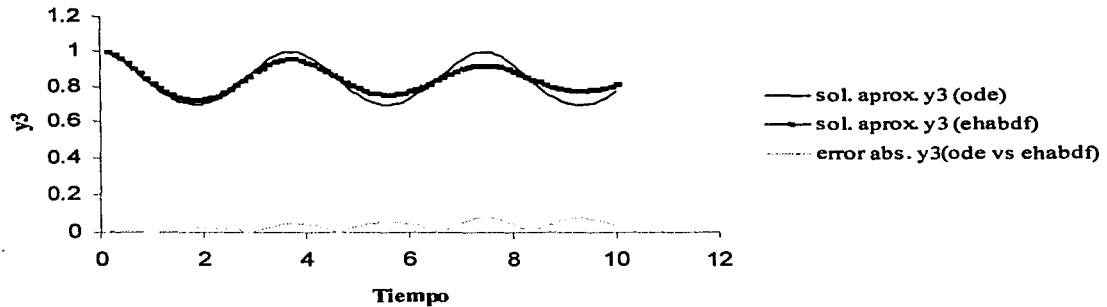
Gráfica 2.6.6.3

Mod. 3. Ecuaciones de Euler, $h=0.125$ s.



Gráfica 2.6.6.4

Mod. 3. Ecuaciones de Euler, $h=0.125$ s.



Gráfica 2.6.6.5

II.6.6.2 Costo computacional

El costo computacional realizado por EHABDF para las 3 simulaciones realizadas se resume en la tabla 2.6.6.2:

Paso De Integración	Matrices Jacobianas calculadas en toda la simulación	Tiempos en donde se cálculo la Matriz Jacobiana.	No. Evaluaciones realizadas al modelo al inicio de la simulación.	No. Evaluaciones realizadas al modelo en cada paso de integración posterior al paso inicial de la simulación
0.125	1	0.125	5	2
0.25	1	0.25	5	2
0.5	1	0.5	5	3 y 2

Tabla 2.6.6.2 Costo computacional cometido en la solución del modelo 3

Nuevamente la matriz jacobiana solamente se cálculo una vez en cada una de las simulaciones efectuadas, el cálculo se realizó al primer paso de integración que se avanzó la solución del modelo. Esta matriz pudo mantenerse constante a lo largo de la simulación por lo cual no hubo necesidad de calcular otra matriz jacobiana.

En relación al número de evaluaciones realizadas al modelo, en cada una de las simulaciones efectuadas, en el primer paso tomado, el modelo se evaluó 5 veces, debido a que en éste paso es necesario calcular la matriz jacobiana, para ello se necesitan tantas evaluaciones del modelo como ecuaciones a resolver, en este caso son 3, una vez calculada la matriz jacobiana, se requieren realizar las iteraciones para obtener la solución del modelo, las cuales, están forzadas por el método de ser al menos 2, por lo que se observa que en éste paso de integración se alcanzó convergencia en 2 iteraciones. Para el resto de los pasos de integración se requirió de 2 ó 3 evaluaciones del sistema, es decir, que en 2 ó 3 iteraciones se alcanzó la convergencia de la solución.

II.6.7 Resumen de los resultados de la validación inicial

En los 3 modelos resueltos por el código EHABDF se pudieron apreciar los siguientes puntos:

- La aproximación numérica obtenida en la resolución de los modelos (1) y (2) con paso de integración de 0.125 s conserva las tendencias cualitativas del modelo.
- La aproximación numérica se mejora al disminuir el paso de integración como se observa en la resolución del modelo (2) con un paso de integración de 0.0125 s.
- La aproximación numérica obtenida en la resolución de los modelos (1) y (2) con los pasos de integración mayores a 0.125 es menos exacta.
- El código EHABDF no funciona bien en la obtención de la solución cuando los valores característicos asociados al sistema de ecuaciones se encuentran fuera de la zona de estabilidad de la fórmula BDF de orden 1, como en el caso del modelo (3).
- EHABDF realiza el menor número posible de evaluaciones del modelo, así como el menor número posible de cálculo de matrices jacobianas.
- El código EHABDF pudo soportar la discontinuidad de la función externa f , sin tener que calcular otra matriz jacobiana.
- No requiere del valor de inicio de las derivadas del modelo, solo necesita del valor de inicio de las variables de estado.

CAPÍTULO III

VALIDACIÓN DEL CÓDIGO EHABDF PARA LA SOLUCIÓN NUMÉRICA DEL SEAD EN TIEMPO REAL

En este capítulo se presentan los resultados de la validación del código EHABDF, al obtener la solución en tiempo real de un SEAD que consta de 161 ecuaciones, 27 de ellas ecuaciones diferenciales y el resto ecuaciones algebraicas. Este SEAD representa el modelo matemático de una planta termoeléctrica desarrollado originalmente por Usoro [Usoro] y adaptado posteriormente en el Instituto de Investigaciones Eléctricas. El modelo representa una unidad de 400 Mega Watts (MW) de potencia.

III.1 Características Principales del Modelo Simplificado de la Planta

III.1.1 Sistema basado en módulos

El modelo de la planta termoeléctrica es un sistema computacional con una estructura basada en módulos, que se resuelve con un método de solución acoplada (ver I.3.1 y I.3.1.2). El modelo se encuentra dividido en 23 módulos, cada uno de los cuales modela una parte de los procesos involucrados en la generación de la energía eléctrica.

Cada módulo está formado por una sola subrutina escrita en lenguaje FORTRAN, dividida en dos secciones por una cláusula condicional if-then-else. La primera sección de la subrutina contiene al conjunto de ecuaciones algebraicas y la segunda sección contiene al conjunto de ecuaciones diferenciales. Para resolver el SEAD completo del modelo, un programa ejecutivo llama al código del método integrador (Runge-Kutta de 4° orden), quien a su vez llama a una subrutina que coordina la secuencia de ejecución de las subrutinas que conforman los módulos. Esta subrutina coordinadora hace una primera llamada a cada una de las 23 subrutinas y evalúa solamente las ecuaciones algebraicas (primera sección de cada subrutina), al finalizar la evaluación de las variables algebraicas, vuelve a llamar a las 23 subrutinas y entonces evalúa las derivadas del modelo (segunda sección de las subrutinas). Una vez evaluadas las ecuaciones diferenciales del modelo el método integrador las resuelve y la solución se manda al programa ejecutivo.

A continuación en la fig. 3.1 se presenta el esquema de computación y el nombre del programa de inicio, el programa ejecutivo y el programa del método integrador del modelo de la planta.

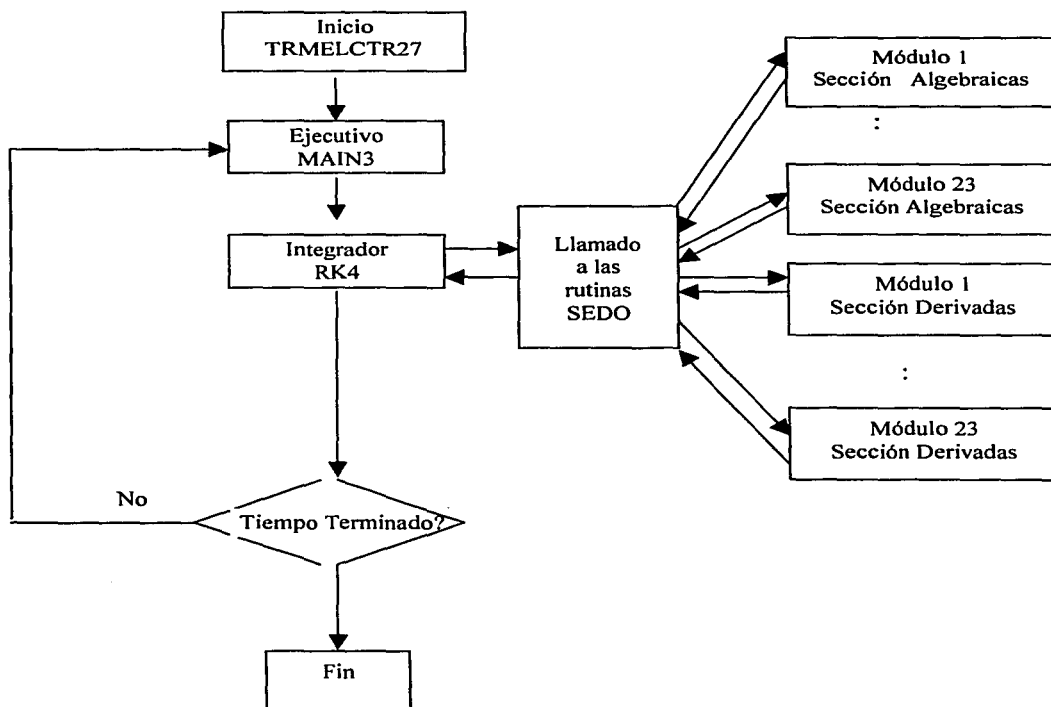


Fig. 3.1. Secuencia de computación del modelo de la planta de potencia

En la tabla 3.1.1 se presentan las 23 subrutinas que conforman a los 23 módulos, y se indica el orden de ejecución de las mismas (considerando a la sección de algebraicas y la sección de derivadas) y una breve descripción del proceso que simulan.

Subrutina	Módulo que la contiene	Secuencia de ejecución en el cálculo de las variables		Proceso Simulado
		Algebraicas	Diferenciales	
CONTSSO	CONTSSO	1	1	Control de la temperatura en el sobrecalentador.
CONTRB	CONTRB	2	2	Control de la turbina.
CONCOMB	CONCOMB	3	3	Control de la combustión.
CONTRHO	CONTRHO	4	4	Control del Hogar
HOGAR	HOGAR	5	5	El hogar de la parte de combustión.
GSHTLSRAD	GSHTLSRAD	6	6	La radiación y a las perdidas de calor en el gas.
GAS	GAS	7	7	La parte del gas de la combustión.
ECNMDR	ECNMDR	8	8	Economizador.
DOMO	DOMO	9	9	Domo, tubería de bajada y la pared de agua, tomándolo como si fuera un solo elemento o módulo.
PRDH2O	PRDH2O	10	10	Pared de agua en su parte de combustión.
SPRHTR	SPRHTR	11	11	Sobrecalentador y el atemperador.
STMFLW	STMFLW	12	12	Control de salida de vapor del sobrecalentador.
DRMWTRFL	DRMWTRFL	13	13	Pared de agua
RHTR	RHTR	14	14	Recalentador en la planta.
HPTRB	HPTRB	15	15	Turbina de alta presión.
CNDNSR	CNDNSR	16	16	Condensador.
LPTRB	LPTRB	17	17	Turbina de baja presión, junto con varios dispositivos mas, que son tomados como un solo elemento o módulo.
TRBGNRTR	TRBGNRTR	18	18	Turbina-generador.
CNDPMP	CNDPMP	19	19	Bomba del condensador.
LPFDHTR	LPFDHTR	20	20	Calentador de agua de alimentación de la turbina de baja presión
DRDR	DRDR	21	21	Deaereador.

Subrutina	Módulo que la contiene	Secuencia de ejecución en el cálculo de las variables		Proceso Simulado
		Algebraicas	Diferenciales	
HPHTR	HPHTR	22	22	Calentador de agua de alimentación a la turbina de alta presión.
BLRFDMP	BLRFDMP	23	23	Bomba de alimentación a la caldera.

Tabla 3.1.1. Subrutinas que conforman los 23 módulos en que se divide el modelo de la planta.

III.1.2 Ecuaciones algebraico-diferenciales explícitas

Las 161 ecuaciones algebraico-diferenciales del modelo se encuentran escritas en forma explícita.

III.1.2.1 Ecuaciones diferenciales

Las ecuaciones diferenciales del SEAD son 27, todas son ecuaciones diferenciales de primer orden. En la tabla 3.1.2 se dan los nombres de las variables de estado correspondientes a estas ecuaciones diferenciales, el nombre de las derivadas y una breve descripción de la variable de estado.

VAR. ESTADO	DERIVADA	DESCRIPCIÓN VARIABLE DE ESTADO
X1SC	DX1SC	AUXILIAR DEL CONTROL DE TEMPERATURA DEL SOBRECALENTADOR.
X2SC	DX2SC	FLUJO DE ATEMPERACION DEL SOBRECALENTADOR (CONTROL).
X1TR	DX1TR	AUXILIAR DEL CONTROL DE LA TURBINA.
X2TR	DX2TR	AUXILIAR DEL CONTROL DE LA TURBINA.
X3TR	DX3TR	APERTURA DE LA VALVULA DE CONTROL (CONTROL).
X1DM	DX1DM	AUXILIAR DEL CONTROL DE DEMANDA DE CARGA.
X1AR	DX1AR	AUXILIAR DEL CONTROL DE FLUJO DE AIRE.
X2AR	DX2AR	FLUJO DE AIRE (CONTROL).
X1CB	DX1CB	AUXILIAR DEL CONTROL DE FLUJO DE COMBUSTIBLE.
X2CB	DX2CB	FLUJO DE COMBUSTIBLE (CONTROL).
X1GR	DX1GR	AUXILIAR DEL CONTROL DE FLUJO DE GAS DE RECIRCULACION.

VAR. ESTADO	DERIVADA	DESCRIPCIÓN VARIABLE DE ESTADO
X2GR	DX2GR	FLUJO DE GAS DE RECIRCULACION (CONTROL).
X1RC	DX1RC	AUXILIAR DEL CONTROL DE LA TEMPERATURA DEL RECALENTADOR.
X2RC	DX2RC	INCLINACION DE QUEMADORES (CONTROL).
X3RC	DX3RC	FLUJO DE ATEMPERACION DEL RECALENTADOR (CONTROL).
HECO	DHECO	ENTALPIA DEL ECONOMIZADOR.
RDRS	DRDRS	DENSIDAD DEL VAPOR EN EL DOMO.
TWWM	DTWWM	TEMPERATURA DE LA PARED DE AGUA (METAL).
RSSO	DRSSO	DENSIDAD EN EL SOBRECALENTADOR.
HSSO	DHSSO	ENTALPIA EN EL SOBRECALENTADOR.
RRHO	DRRHO	DENSIDAD EN EL RECALENTADOR.
HRHO	DHRHO	ENTALPIA EN EL RECALENTADOR.
NTR	DNTR	VELOCIDAD DE LA TURBINA.
DELTA	DDELTA	ANGULO DE POTENCIA.
HIHO	DHIHO	ENTALPIA DEL CALENTADOR DE BAJA PRESION.
HDEW	DHDEW	ENTALPIA DEL AGUA DEL DESAERADOR.
HHHO	DHHHO	ENTALPIA CALENTADOR DE ALTA PRESION.

Tabla 3.1.2. Variables de estado del modelo.

III.1.2.2 Ecuaciones algebraicas

La mayoría de las ecuaciones algebraicas sirven para realizar cálculos auxiliares con los cuales se evalúan los valores de las derivadas; sin embargo, existen 42 variables algebraicas que son salidas a los tableros del simulador, por lo que son las que se consideran únicamente como las algebraicas del SEAD. La tabla 3.1.3 contiene el nombre de estas 42 variables.

VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA
WG	HDRW	WFW	WFL	TRHO	HFPO
TFNI	RDRW	WSSO	XGG	WHPO	HCPO
TWWGO	WDRS	WSY	QSH	HHPO	MWGN
QWWGM	PSSO	ACV	QRH	THPO	HLHD
HDRS	SSSO	WRY	QEC	MWHP	HDED
TDRS	TSSO	WAR	PHPO	WRHO	HHHD
PDRS	QWWMW	WGR	PRHO	MWLP	MWTRO

Tabla 3.1.3. Variables algebraicas de salida a tableros del simulador

III.1.2.3 Limitación de variables del SEAD

En la formulación matemática del modelo existen expresiones que verifican que la solución obtenida en 15 ecuaciones diferenciales ordinarias (variables de estado) y en 7 variables algebraicas, se encuentren dentro de un rango dado.

A este tipo de expresiones, se les denominó en este trabajo ecuaciones limitadoras o simplemente limitadores. Estas ecuaciones trabajan con base en un límite inferior y un límite superior dentro de los cuales debe encontrarse el valor de la variable a la cual verifican. Estas expresiones modifican el valor de la variable, únicamente cuando éste no se encuentra dentro de los límites dados, en cuyo caso el valor de la variable es cambiado por el valor del límite permisible más cercano.

Estas ecuaciones limitadoras representan un problema potencial para la utilización del código EHABDF, ya que ellas pueden modificar el valor obtenido por el código EHABDF de 15 variables de estado y 7 variables algebraicas, si su valor no se encuentra dentro de los rangos permisibles de operación, modificando con ello, el punto de partida de la solución para el siguiente paso de integración que tome EHABDF.

En la tabla 3.1.4 se presentan las 15 variables de estado y las 7 variables algebraicas que son salidas a los tableros del simulador y que están siendo limitadas.

VARIABLE	TIPO	LIM. INF.	LIM. SUP.	VARIABLE	TIPO VAR.	LIM. INF.	LIM. SUP.
X1SC	Estado	1.0	5.0	X2GR *	Estado	1.0	KCWGR≤5.0
X2SC	Estado	1.0	5.0	X1RC	Estado	1.0	5.0
X1TR	Estado	-1.0	1.0	X2RC	Estado	1.0	5.0
X2TR	Estado	0.0	5.0	X3RC	Estado	1.0	5.0
X3TR	Estado	1.0	5.0	WSY	Algebraica	0.0	100.0
X1DM	Estado	1.0	5.0	ACV	Algebraica	0.0	1.0
X1AR	Estado	1.0	5.0	WAR	Algebraica	0.0	1381.4
X2AR	Estado	1.0	5.0	WFL	Algebraica	0.0	90.0
X1CB	Estado	1.0	5.0	WGR	Algebraica	0.0	500.0
X2CB	Estado	1.0	5.0	WRV	Algebraica	0.0	50.0
X1GR	Estado	1.0	5.0	XGG	Algebraica	-0.5236	0.5236

Tabla 3.1.4

Como puede verse en la tabla 3.1.4 los límites permisibles para estas variables son límites constantes. Sin embargo, para la variable de estado X2GR, existe además, otra expresión dentro del modelo, que verifica si esta variable es mayor que la variable algebraica KCWGR, si esto es así, entonces el valor de la variable de estado X2GR es cambiado por el valor de esa algebraica, en caso contrario, el valor de la variable de estado es obtenido a través de la solución de la derivada DX2GR.

Esta asignación de la variable de estado a través de una variable algebraica también representa un problema al código EHABDF, debido a que se modifica el valor de la variable de estado directamente y no a través de la derivada correspondiente.

III.1.3 Secuencia de ejecución de las ecuaciones

La secuencia de ejecución del SEAD del modelo se obtiene de acuerdo al orden en que se ejecutan los módulos y de acuerdo al orden en el cual se evalúan las ecuaciones dentro de cada módulo.

Con el propósito de saber si el modelo tenía una correcta secuencia de ejecución de sus ecuaciones o bien si requería de valores de inicio en variables no inicializadas, se aplicó al modelo una prueba conocida como Prueba de Consistencia [Ojeda_2]. La prueba de consistencia tiene por objetivo detectar si existen variables mal inicializadas, es decir, que o bien les falta un valor de inicio no dado (ecuaciones no lineales) o que se encuentran en un orden de ejecución equivocado (ecuaciones lineales). La prueba consiste en hacer dos evaluaciones del modelo en un mismo tiempo dado, la primera vez se evalúa con el conjunto inicial de datos que se tienen del modelo en dicho tiempo y la segunda vez se evalúa el modelo con los valores obtenidos de la primera evaluación, realizando posteriormente una comparación entre los resultados obtenidos en las dos evaluaciones. Si la diferencia no es nula entonces se tiene un problema de inconsistencia.

III.1.3.1 Problemas encontrados en la secuencia del SEAD

La aplicación de la prueba de consistencia al modelo mostró que éste tenía problema de inconsistencia (secuencia incorrecta de ejecución de las ecuaciones y/o variables sin un valor inicial requerido). Debido a ello, se decidió obtener las dependencias y secuencia de ejecución de las ecuaciones del modelo, así como las variables de entrada y de salida a cada módulo, para detectar con ello las posibles inconsistencias del modelo. En el apéndice A, sección A.1 se presenta en la tabla a.1 la secuencia de ejecución así como las dependencias de las 161 ecuaciones del modelo.

Con los resultados obtenidos de las dependencias de las ecuaciones junto con su secuencia de ejecución, se formó la Matriz Estructural del Modelo [Ojeda_2], esta matriz contiene en forma binaria la estructura de las dependencias de las ecuaciones entre sí, de la siguiente forma:

$$c_{i,j} = \begin{cases} 1, & \text{si la ecuación } i \text{ depende de la ecuación } j \\ 0, & \text{en caso contrario} \end{cases}$$

El orden en que se ejecutan las ecuaciones dentro del modelo es el orden en que aparecen en los renglones de la matriz.

Cuando la matriz estructural del modelo resulta ser una matriz con una estructura triangular inferior, entonces se tiene una correcta secuencia de ejecución de las ecuaciones y además se encuentra libre de ciclos iterativos. En caso contrario, puede ser que algunas ecuaciones se encuentren dentro de ciclos iterativos o que algunas ecuaciones se encuentran en un orden de ejecución inadecuado.

En el apéndice B se presenta la matriz estructural inicial obtenida del modelo, como puede verse, ésta muestra la existencia de ecuaciones que tienen un orden de ejecución inadecuado ya que dependen de variables que no han sido calculadas y que son calculadas posteriormente; al revisar estas variables, se detectó que no cuentan con un valor de inicio, esto hace que las ecuaciones sean evaluadas incorrectamente al inicio de la simulación y que en el transcurso de la simulación sean evaluadas con información atrasada. A continuación presentamos las ecuaciones encontradas.

# ec	Tipo de ecuación	Variable Modificada	Tipo Var	Dependencia calculada posteriormente y sin valor de inicio
3	LIMITADOR	KTSSO	ALG	WSSO
5	TRANSDUCTOR	CTSSO	ALG	TSSO
12	TRANSDUCTOR	CMWTRO	ALG	MWTRO
27	TRANSDUCTOR	CPSSO	ALG	PSSO
50	TRANSDUCTOR	CTRHO	ALG	TRHO

Tabla 3.1.4 Ecuaciones con un orden de ejecución incorrecto

Debido a esta característica, al modelo original se le llamó modelo desordenado explícito ya que las ecuaciones se encuentran escritas en forma explícita y con problemas de ordenamiento.

III.1.3.2 Ordenamiento de la secuencia de ejecución de las ecuaciones

Los problemas de la secuencia de ejecución de las ecuaciones del modelo mencionados en la sección III.1.3.1, presentan un problema potencial para la resolución del modelo a través

del código EHABDF, debido a que las ecuaciones algebraicas auxiliares que se utilizan para la evaluación de las derivadas o de las variables algebraicas que salen a los tableros del simulador, no deben resolverse en forma simultánea, sin embargo, deben tener una secuencia de ejecución correcta, para evitar retrasos de información en la resolución del modelo. Debido a esto, se decidió ordenar la secuencia de ejecución de las ecuaciones del modelo.

Para realizar el ordenamiento de las ecuaciones del modelo, se aplicó el algoritmo de ordenamiento denominado Algoritmo de Ledet [Ojeda_2]. Este algoritmo trabaja a partir de la matriz estructural inicial obtenida del modelo y con base en algoritmos de teoría de gráficas busca minimizar los ciclos iterativos de las ecuaciones presentes en el modelo, proporcionando como resultado una matriz estructural final, triangular inferior por bloques, que contiene el ordenamiento óptimo de las ecuaciones del modelo.

La matriz estructural final reportada por el algoritmo de Ledet del modelo que nos ocupa, ver apéndice D es una matriz triangular inferior, libre de bloques (ciclos iterativos).

La implantación de este nuevo ordenamiento obtenido por el algoritmo de Ledet, requería de varios cambios de posición de las ecuaciones del modelo y de reestructuración de subrutinas, debido a que algunas ecuaciones requerían ser migradas de una subrutina a otra. Para minimizar el número de cambios y con ello la respuesta física del proceso que se modela, se decidió realizar únicamente los cambios de orden estrictamente necesarios. Esto es, modificar únicamente el orden de las ecuaciones cuyas dependencias se encuentran marcadas con 1 fuera de la diagonal en la matriz estructural inicial, y las demás dejarlas en el orden original.

De esta manera, solo se tuvo que cambiar el orden local de ejecución de dichas ecuaciones dentro de los 4 primeros módulos del modelo:

1.-Contsso	2.- Contrb	3.- Concomb	4.- Contrho
------------	------------	-------------	-------------

Las ecuaciones cambiaron de posición dentro de los mismos módulos, por lo que al no transferirse ecuaciones de un módulo a otro, no se requirió cambiar la secuencia de ejecución de los módulos. Las entradas y salidas a cada módulo tampoco sufrieron cambios. La tabla con el nuevo ordenamiento obtenido con estos cambios del modelo se presenta en el apéndice C y su matriz estructural final asociada se presenta en el apéndice E. Al modelo ordenado se le denominó modelo ordenado explícito, debido a que sus ecuaciones se encuentran ordenadas y sus expresiones están escritas en forma explícita.

III.2 Pruebas Elegidas para la Validación del Código EHABDF

La cantidad de energía eléctrica generada en una planta de potencia, depende en gran parte de la demanda de energía eléctrica requerida por los usuarios. Esta demanda puede variar a lo largo de un día, algunas veces se mantiene constante y otras veces aumenta o disminuye según las necesidades de los usuarios.

En las plantas de potencia, esta demanda de energía eléctrica se conoce como demanda en la carga.

Cuando la demanda en la carga permanece constante, la cantidad de energía producida por la planta permanece constante, por lo que las variables involucradas en el proceso de generación de la energía permanecen sin cambio alguno y se dice que la planta se encuentra en un estado estacionario o estable. Este estado estable se traduce en el modelo de la planta cuando las derivadas presentes en el modelo son iguales a cero (tendiendo a cero).

Cuando la demanda en la carga aumenta o disminuye, la cantidad de energía eléctrica que la planta produce también requiere ser aumentada o disminuida según el caso. Estas variaciones de la demanda de energía hacen que las variables involucradas en los procesos de generación de energía eléctrica también varíen y se dice que la planta se encuentra en un estado transitorio. La demanda en la carga se traduce en los modelos como una función externa al modelo, la cual perturba el valor de la derivada para hacerla salir del estado estable, si la demanda está cambiando de valor, la derivada dejará de ser igual a cero, por lo que el valor de la variable de estado involucrada cambia.

Para la validación de los resultados del código EHABDF en la simulación del modelo ordenado de la planta termoeléctrica de 27 ecuaciones diferenciales se eligieron dos pruebas, una correspondiente a la simulación de un estado estable y la otra de un estado transitorio.

III.2.1 Prueba No 1. Modelo en estado estacionario

Esta prueba consiste en simular el modelo durante un período de 400 segundos, con una demanda en la carga inicial del 100% la cual permanece constante durante el período de simulación.

La prueba se inicia con un conjunto de valores iniciales dados para el 100% de carga. Como se indicó en la sección anterior las derivadas del modelo son todas iguales a cero, durante la simulación el modelo no debe presentar ninguna perturbación, es decir la solución del SEAD permanece constante.

III.2.2 Prueba No. 2. Modelo en estado transitorio

Esta prueba consiste en simular el modelo de la planta termoeléctrica en un estado transitorio, es decir, se inicia la simulación con un conjunto de datos dado y después de un período de simulación se mete una perturbación al modelo que lo hace cambiar de su estado estable, en este caso las derivadas del modelo empiezan a ser distintas de cero. El transitorio elegido consiste en disminuir la función demanda del 100% de carga al 77.5% de carga a una razón del 15% por minuto. La simulación inicial para introducir este estado transitorio consiste en simular un estado estable al 100% de carga durante 10 seg.; posteriormente se mete la perturbación al modelo para alcanzar la demanda en la carga al valor requerido de 77.5%, una vez obtenida permanecerá constante a partir de ese momento hasta alcanzar los 400 segundos de simulación del modelo.

III.2.3 Período de muestro de las variables del modelo

En la aplicación de las pruebas descritas anteriormente para la validación del código EHABDF, se eligió un período de muestreo de 1 segundo para las variables del modelo, iniciando el muestreo a partir del primer segundo que EHABDF alcance exitosamente la simulación del modelo.

III.3 Generación de una Base de Referencia como Solución Exacta para la Aplicación de las Pruebas

Para la validación del código EHABDF, se seleccionó el modelo ordenado explícito que elimina los problemas de retraso en la información, que se mencionan en las secciones III.1.3.1 y III.1.3.2, y será a este modelo al que se le aplicarán las pruebas descritas en la sección III.2.

Como no se cuenta con datos de la solución exacta del modelo, para poder realizar la comparación de los resultados obtenidos por el código EHABDF, se generó una base de referencia para la aplicación de las pruebas al modelo ordenado y resuelto a través del código EHABDF, ésta consiste de los resultados obtenidos en la aplicación de las pruebas al modelo ordenado explícito resuelto a través del método integrador Runge-Kutta de cuarto orden con paso de integración de 0.1 seg., ya que con este método y paso de integración el modelo original de la planta fue validado con datos de una planta real [Uso] en la aplicación de las mismas pruebas.

Para establecer esta base de referencia, el modelo ordenado explícito se validó respecto al modelo desordenado explícito, con los siguientes pasos:

- 1.- Se le aplicó la prueba de consistencia al modelo ordenado utilizando la prueba no. 2, ver sección III.2.2, en los siguientes tiempos: 0, 5, 15, 30, 45, 60, 75, 90, 95, 110, 120, 130, 140, 150, 200, 380 s. El resultado de la prueba fue satisfactorio en cada tiempo en el que fue efectuada, con esto se tiene la confianza de que la nueva secuencia de las ecuaciones del modelo es correcta.

- 4.- Se compararon los resultados obtenidos con el modelo ordenado contra los resultados obtenidos por el modelo original, midiendo la diferencia cometida por el modelo ordenado respecto del modelo original, ésta diferencia se midió como error relativo si el valor absoluto de la variable reportado por el modelo original era mayor que 1 y como error absoluto cuando el valor absoluto de la variable era menor o igual a 1. En el apéndice F se presentan los máximos errores cometidos en las 27 variables de estado y en las 42 variables algebraicas en la aplicación de las pruebas (tablas f.3.1 y f.3.2 respectivamente). En la tabla f.3.1 se aprecia que en las variables de estado, el error más grande que se cometió en la prueba del modelo en estado estacionario se cometió en la variable X2AR con un error de 0.1769 % y que en la prueba del modelo en estado transitorio, el error más grande se cometió en la variable X2CB con un error de 0.3101%. En la tabla f.3.2 se aprecia que en las variables algebraicas de salida a los tableros del simulador el error más grande obtenido en la prueba del estado estacionario se obtuvo en la variable QEC con un error del 0.2633% y para la prueba del modelo en estado transitorio se obtuvo en la variable QSH con un error de 0.7728%.

III.4 Solución de las Ecuaciones Algebraico-Diferenciales del Modelo Ordenado por Medio de EHABDF

Para la aplicación de las pruebas descritas en la sección III.2 con el modelo ordenado utilizando el código EHABDF, se debe resolver en forma simultánea el sistema de ecuaciones algebraico-diferencial que consta de las 27 ecuaciones diferenciales ordinarias y de las 42 variables algebraicas que son salidas a los tableros del simulador, el resto de las variables algebraicas no se deben resolver en forma simultánea, ya que como se mencionó en la sección III.1.2.2, sólo son variables auxiliares que sirven para guardar cálculos auxiliares, que se utilizan para la evaluación de las derivadas del modelo o de las variables algebraicas que son salidas a los tableros del simulador.

Los parámetros de simulación que se utilizan para la solución del modelo en la aplicación de las pruebas, son los siguientes:

Paso de integración (h)	Error de tolerancia	No. máximo de iteraciones	Cálculo De Jacobiano	Épsilon de la máquina.
0.1 seg.	0.33	4	Cuando EHABDF lo requiera.	1.192093E-07

Tabla 3.4.1

III.4.1 Adaptaciones realizadas al modelo

Para poder resolver el modelo a través de EHABDF se necesitaron realizar modificaciones al modelo ordenado. En las siguientes secciones se presentan estas modificaciones.

III.4.1.1 Ecuaciones diferenciales y algebraicas en forma implícita

Las 27 ecuaciones diferenciales ordinarias del modelo así como las 42 variables algebraicas que son salidas a los tableros del simulador se cambiaron de su forma explícita a una forma residual, ésta forma residual consiste en guardar en un residuo la evaluación de la ecuación original moviendo el lado izquierdo de la ecuación con signo negativo al lado derecho de la ecuación y en el lado izquierdo de la ecuación se asignó a un residuo el valor de la ecuación implícita resultante.

III.4.1.2 Limitadores

Como se mencionó en la sección III.1.2.3 y en la tabla 3.1.4, dentro del modelo se limitan 15 variables de estado y 7 variables algebraicas (salidas a los tableros del simulador), estas limitaciones se realizan a través de los limitadores. Debido a que los limitadores de estas variables modifican su valor, representan un problema al código EHABDF, pues éste requiere que no se modifiquen los valores de las variables calculadas por él. Estos limitadores no se pueden simplemente eliminar del modelo, ya que EHABDF puede obtener valores de dichas variables que no se encuentren dentro del rango de operación válido, tampoco pueden permanecer dentro del modelo porque modifican externamente a EHABDF el valor de las variables. Sin embargo, debido a que los límites contra los cuales se comparan las variables son constantes y por ende no dependen de ninguna variable adicional del modelo, estos limitadores se eliminaron de las ecuaciones del modelo y se implementaron dentro de una nueva subrutina llamada `lim_est`, la cual se creó para que el código EHABDF pueda limitar por el mismo las variables calculadas por él, una vez que estas han sido calculadas y así trabajar con valores dentro de los rangos permisibles de operación.

Además de estos limitadores, se mencionó en la sección III.1.2.3, que la variable de estado X2GR se encuentra también limitada a través de la variable algebraica KCWGR, ya que, cuando la variable de estado X2GR tiene un valor mayor al de la variable algebraica, el valor de X2GR es cambiado por el valor de esta variable algebraica.

Para solucionar este problema, dentro del modelo se eliminó la asignación directa de la variable de estado al convertirla en una ecuación implícita, cuyo residuo correspondiente tiene la misma identificación que el residuo de la ecuación diferencial implícita DX2GR. De esta forma se verifica si el valor de la variable de estado X2GR es mayor al de la variable algebraica, si esto es así, entonces el residuo que se evalúa es la diferencia entre la variable de estado y la variable algebraica. En caso contrario, el residuo que se evalúa es el correspondiente a la ecuación diferencial DX2GR escrita en forma implícita.

III.4.1.3 Valores iniciales

Para poder resolver las 27 variables de estado y las 42 variables algebraicas que son salidas a los tableros del simulador, se necesitan valores de inicio de todas estas variables, sin embargo, el modelo original solo dispone del valor de inicio de las 27 variables de estado y de algunas variables algebraicas. Por este motivo se tomaron como valores de inicio de las 27 variables de estado y de las 42 variables algebraicas los valores obtenidos en el primer segundo de simulación en la aplicación de la prueba del modelo en estado estacionario con el modelo ordenado y resuelto a través del método de Runge-Kutta de 4° orden (Base de referencia).

III.4.2 Adaptaciones realizadas al código EHABDF para la aplicación de las pruebas

El código EHABDF se modificó con la implantación de la subrutina `lim_est` para que pueda trabajar con los valores permisibles de las 15 variables de estado y de las 7 variables algebraicas limitadas mencionadas en la sección III.4.1.2 al momento de obtener la solución del SEAD.

La subrutina `lim_est` es llamada por EHABDF inmediatamente después de:

- Predecir el valor de la solución que se busca.
- Perturbar las variables para realizar el cálculo del jacobiano.
- Actualizar el valor obtenido de la solución en cada iteración que se realiza.

III.5 Resultados Obtenidos

Una vez realizadas las modificaciones requeridas al modelo ordenado y las adaptaciones necesarias al código EHABDF, se aplicó la prueba del modelo en estado estacionario pero el código EHABDF no pudo avanzar ni un solo paso de integración la solución del modelo, debido a que la matriz jacobiana que calculó al inicio de la simulación resulto singular.

Al analizar la matriz jacobiana calculada por EHABDF se observaron dos columnas de la matriz cuyos elementos son únicamente ceros, por tanto es una matriz singular.

Estas columnas corresponden a dos variables algebraicas limitadas y cuyos valores de inicio son iguales a sus límites superiores:

Variable Algebraica	Limite Superior	Valor Inicial
XGG	0.5236	0.5236
WSY	100.0	100.0

Tabla 3.5.3

Cuando EHABDF perturba estas variables para calcular las derivadas parciales de las ecuaciones del SEAD respecto a estas variables, se incrementa su valor y sale del rango permisible de operación, por lo que es limitado el valor al límite superior, con ello, el valor perturbado de estas variables coincide con el valor sin perturbar, ya que EHABDF limita las ecuaciones una vez que ha realizado la perturbación, esto hace que la evaluación del modelo en la variable perturbada coincida con la evaluación del modelo en la variable sin perturbar, lo que hace que las diferencias de estas evaluaciones sean cero y con ello las derivadas parciales de estas variables y los elementos de la matriz sean ceros también.

III.6 Forma Alternativa de Solucionar el Modelo Ordenado por Medio de EHABDF

Debido al mal condicionamiento de la matriz jacobiana obtenidas por EHABDF en la solución simultánea del SEAD del modelo, y que el problema se presentaba con las ecuaciones algebraicas, se decidió resolver únicamente en forma simultánea las 27 ecuaciones diferenciales del modelo y resolver en forma secuencial las ecuaciones algebraicas.

Los parámetros de simulación requeridos por EHABDF en la solución del modelo en la aplicación de las pruebas, permanecieron sin cambio (ver tabla 3.4.1).

III.6.1 Adaptaciones realizadas al modelo para la aplicación de las pruebas

Para poder probar al código EHABDF en la resolución simultánea de las 27 ecuaciones diferenciales se requirieron hacer nuevas adaptaciones al modelo ordenado explícito. En este caso, solo se hicieron modificaciones a las ecuaciones diferenciales, quedando las ecuaciones algebraicas escritas en forma explícita, el orden de las ecuaciones del modelo ordenado tampoco sufrió cambio alguno. El modelo resultante con las adaptaciones realizadas, recibió el nombre de modelo ordenado diferencial implícito.

III.6.1.1 Ecuaciones diferenciales en forma implícita

Las 27 ecuaciones diferenciales escritas en forma explícita del modelo ordenado se cambiaron a la forma residual, la forma residual consiste en guardar en un residuo la evaluación de la ecuación diferencial escrita en forma implícita.

III.6.1.2 Limitadores de variables de estado

La limitación de las 15 variables de estado, ver sección III.4.1.2, fue eliminada del modelo, en su lugar se implantaron en una nueva subrutina nombrada `lim_est`, la cual es llamada por el código EHABDF para que pueda hacer internamente las limitaciones de las variables de estado.

Además de estos limitadores, como se mencionó en la sección III.1.2.3, para la limitación adicional de la variable de estado X2GR; se realizó lo siguiente:

- Se eliminó la asignación directa de la variable de estado, convirtiéndola en una ecuación implícita, cuyo residuo correspondiente tiene la misma identificación que el residuo de la ecuación diferencial implícita DX2GR.
- Se programaron instrucciones dentro del modelo para verificar si el valor de la variable de estado X2GR es mayor al de la variable algebraica, si esta es así, entonces el residuo que se evalúa es la diferencia entre la variable de estado y la variable algebraica; en caso contrario, el residuo que se evalúa es el correspondiente a la ecuación diferencial DX2GR escrita en forma implícita.

III.6.2 Adaptaciones realizadas al código EHABDF para la aplicación de las pruebas

Para que el código EHABDF pueda trabajar con los valores permisibles de las 15 variables de estado mencionadas en la sección III.6.1.2, se modificó el código original de EHABDF con la implementación de la subrutina `lim_est`, la cual contiene las limitaciones de las 15 variables de estado. Ésta subrutina es llamada por EHABDF inmediatamente después de:

- Predecir el valor de la solución que busca.
- Perturbar las variables para realizar el cálculo del jacobiano.
- Actualizar el valor obtenido de la solución en cada iteración que se realiza.

III.7 Resultados Obtenidos con la Forma Alternativa

A continuación se presentan los resultados obtenidos por el código EHABDF considerando la forma alternativa de solución del modelo ordenado, en la aplicación de las pruebas descritas en la sección III.2.

III.7.1 Aproximación numérica

Se aplicaron las pruebas descritas en la sección III.2 a la forma alternativa de resolver el modelo a través de EHABDF con un paso de integración de 0.1 seg. y se obtuvieron muestras de los resultados obtenidos en cada segundo de simulación de las 27 variables de estado y de las 42 variables algebraicas que son salidas a los tableros del simulador.

Las muestras obtenidas de los valores de las variables citadas, se compararon con las muestras obtenidas por la base de referencia en la aplicación de las mismas pruebas. Se recuerda que la base de referencia es la solución del modelo ordenado explícito integrado a través del método de Runge-Kutta de 4° orden utilizando un paso de integración de 0.1 seg.

La comparación se realizó sacando el máximo error cometido a lo largo de toda la simulación en cada una de las variables de estado y en las variables algebraicas que salen a los tableros del simulador. El error se calculó como error relativo si el valor absoluto de la

variable reportado por el modelo ordenado explícito era mayor que 1 y como error absoluto cuando el valor absoluto obtenido era menor o igual a 1.

A continuación se presenta la tabla 3.7.1 que contiene los máximos errores obtenidos de las 27 variables de estado en la aplicación de las pruebas:

VARIABLE DE ESTADO	PRUEBA 1. ESTACIONARIO		PRUEBA 2. TRANSITORIO	
	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)
X1SC	0.	0.	0.	0.
X2SC	0.	0.	0.	0.
X1TR	0.	0.	0.1177	128.0
X2TR	0.3420E-01	76.00	0.3692E-01	11.00
X3TR	0.3422E-01	72.00	0.3697E-01	12.00
X1DM	0.5032E-01	176.0	0.7238E-01	136.0
X1AR	0.4313E-01	173.0	0.6660E-01	128.0
X2AR	0.5749E-01	94.00	0.2908	96.00
X1CB	0.3835E-01	181.0	0.5692E-01	128.0
X2CB	0.5511E-01	101.0	0.2930	95.00
X1GR	0.	0.	0.	0.
X2GR	0.2003E-02	11.00	4.826	68.00
X1RC	0.	0.	0.	0.
X2RC	0.	0.	0.	0.
X3RC	0.	0.	0.	0.
HECO	0.1925E-01	392.0	0.8827E-02	393.0
RDRS	0.8503E-02	134.0	0.2341E-01	93.00
TWWM	0.	0.	0.8861E-02	156.0
RSSO	0.1829E-01	394.0	0.2204E-01	100.00
HSSO	0.6459E-02	5.000	0.6783E-02	113.0
RRHO	0.4676E-01	117.0	0.3136E-01	16.00
HRHO	0.7038E-02	363.0	0.7289E-02	120.0
NTR	0.	0.	0.	0.
DELTA	0.1465E-01	14.00	0.9135E-01	19.00
HIHO	0.5124E-02	25.00	0.5274E-02	397.0
HDEW	0.1594E-01	388.0	0.2032E-01	378.0
HHHO	0.1346E-01	394.0	0.1624E-01	398.0

Tabla 3.7.1 Máximos errores cometidos en las variables de estado en la validación (EHABDF vs RK4, h=0.1 seg.).

En la tabla 3.7.1 se observa lo siguiente:

- En la prueba del modelo en estado estacionario los errores más grandes de todas las variables de estado, se cometieron en las variables X2AR y X2CB cuyos máximos errores fueron 0.5749E-01% y 0.5511E-01% respectivamente.

- En la prueba del modelo en estado transitorio los errores más grandes de todas las variables de estado, se cometieron en las variables X2GR y X2CB cuyos máximos errores fueron 4.826% y 0.2930% respectivamente.

A continuación se presenta la tabla 3.7.2 que contiene los máximos errores cometidos en las 42 variables algebraicas en la aplicación de las pruebas.

VARIABLE ALGEBRAICA SALIDA A LOS TABLEROS	PRUEBA 1. ESTACIONARIO		PRUEBA 2. TRANSITORIO	
	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)
WG	0.5396E-01	92.00	0.5204	95.00
TFNI	0.1431E-01	154.0	0.2699	77.00
TWWGO	0.1717E-01	159.0	0.2023	77.00
QWWGM	0.6315E-01	105.0	0.5730	77.00
HDRS	0.	0.	0.9228E-02	60.00
TDRS	0.	0.	0.8943E-02	138.0
PDRS	0.3856E-02	8.000	0.1342E-01	111.0
HDRW	0.2705E-02	40.00	0.5641E-02	93.00
RDRW	0.2955E-02	89.00	0.8742E-02	65.00
WDRS	0.4042E-01	85.00	0.5692E-01	85.00
PSSO	0.4145E-02	5.000	0.1433E-01	117.0
SSSO	0.6268E-02	327.0	0.6432E-02	111.0
TSSO	0.1256E-01	395.0	0.1349E-01	128.0
QWWMW	0.5318E-01	8.000	0.2123	81.00
WFW	0.3907E-01	79.00	0.4859E-01	69.00
WSSO	0.3206E-01	86.00	0.3446E-01	11.00
WSY	0.	0.	0.	0.
ACV	0.3422E-01	86.00	0.3819E-01	11.00
WRY	0.	0.	0.	0.
WAR	0.8209E-01	93.00	0.5208	95.00
WGR	0.	0.	0.5191	95.00
WFL	0.7141E-01	95.00	0.5259	94.00
XGG	0.	0.	0.	0.
QSH	0.7541E-01	95.00	1.000	95.00
QRH	0.6229E-01	99.00	0.6370	94.00
QEC	0.9387E-01	97.00	0.3868	77.00
PHPO	0.2994E-01	88.00	0.3244E-01	18.00
PRHO	0.3313E-01	101.0	0.3137E-01	21.00
TRHO	0.1574E-01	390.0	0.8515E-02	102.0
WHPO	0.3184E-01	70.00	0.3319E-01	11.00
HHPO	0.8727E-02	294.0	0.9180E-02	101.0
THPO	0.1993E-01	200.0	0.1480E-01	67.00
MWHP	0.2644E-01	181.0	0.2646E-01	11.00
WRHO	0.3184E-01	113.0	0.3193E-01	14.00
MWLP	0.3577E-01	381.0	0.362E-01	18.00
HFPO	0.1527E-01	388.0	0.1946E-01	378
HCPO	0.	0.	0.	0.
MWGN	0.1303E-01	6.000	0.8097E-01	26.00
HLHD	0.	0.	0.7624E-02	101.0
HDED	0.7126E-02	37.00	0.7218E-02	30.00
HHHD	0.1087E-01	38.00	0.1105E-01	31.00
MWTRO	0.2823E-01	364.0	0.2556E-01	67.00

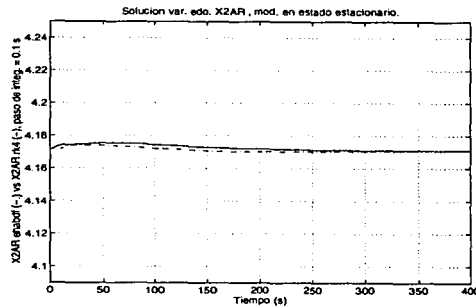
Tabla 3.7.2 Máximos errores cometidos en las variables algebraicas de salida a los tableros, en la validación (EHABDF vs RK4, h=0.1 seg.).

En la tabla 3.7.2 se aprecia lo siguiente:

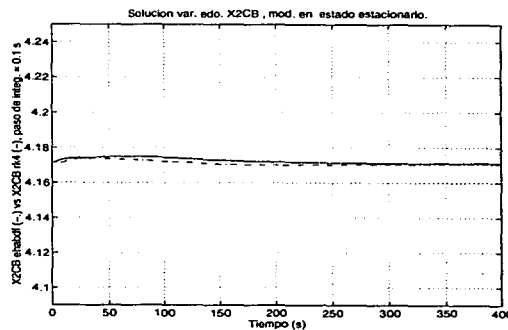
- En la prueba del estado estacionario los errores más grandes cometidos en las 42 variables algebraicas se presentan en las variables QEC y WAR cuyos máximos errores fueron de $0.9387E-01\%$ y $0.8209E-01\%$ respectivamente.
- En la prueba del modelo en estado transitorio los errores más grandes de las variables algebraicas se obtuvieron en las variables QSH y QRH cuyos máximos errores fueron 1.0% y 0.637% respectivamente.

A continuación se presentan las gráficas de algunas de las variables con los errores más grandes obtenidos en la aplicación de las pruebas.

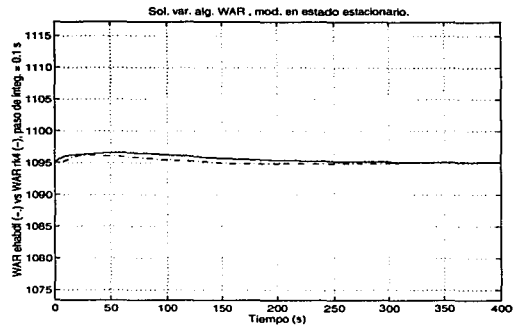
- Estado estacionario



Gráfica 3.7.1 Solución de la variable de estado X2AR, obtenida por el código EHABDF y el código RK4, en la simulación del modelo en estado estacionario. Los valores de la solución han sido muestreados cada segundo.

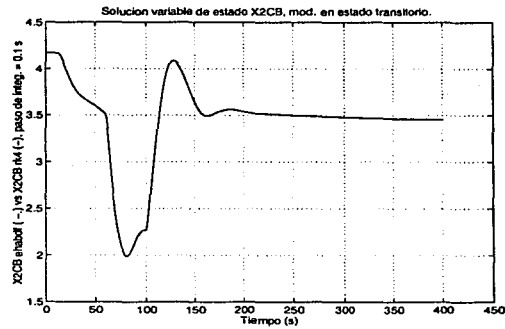


Gráfica 3.7.2 Solución de la variable de estado X2CB, obtenida por el código EHABDF y el código RK4, en la simulación del modelo en estado estacionario. Los valores de la solución han sido muestreados cada segundo.

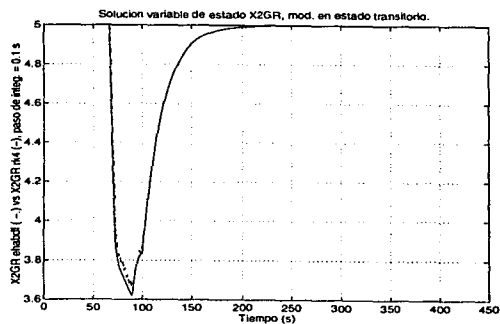


Gráfica 3.7.3 Solución de la variable algebraica WAR, obtenida por el código EHABDF y el código RK4, en la simulación del modelo en estado estacionario. Los valores de la solución han sido muestreados cada segundo.

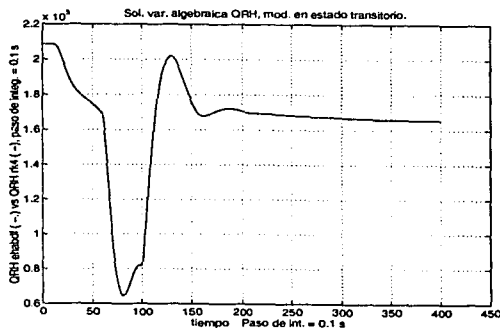
- Estado transitorio



Gráfica 3.7.4 Solución de la variable de estado X2CB, obtenida por el código EHABDF y el código RK4, en la simulación del modelo en estado transitorio. Los valores de la solución han sido muestreados cada segundo.



Gráfica 3.7.5 Solución de la variable de estado X2GR, obtenida por el código EHABDF y el código RK4, en la simulación del modelo en estado transitorio. Los valores de la solución han sido muestreados cada segundo.



Gráfica 3.7.6 Solución de la variable algebraica QRH, obtenida por el código EHABDF y el código RK4, en la simulación del modelo en estado transitorio. Los valores de la solución han sido muestreados cada segundo.

III.7.2 Costo computacional

El costo computacional realizado por el código EHABDF en la aplicación de las pruebas descritas en la sección III.2, se midió a través de:

1. El número de matrices de iteración requeridas por EHABDF.
2. El número de evaluaciones realizadas al modelo por EHABDF.
3. El tiempo de UPC transcurrido en cada paso de integración ($h=0.1$ seg.) para obtener la solución del modelo.
4. Memoria requerida por el código

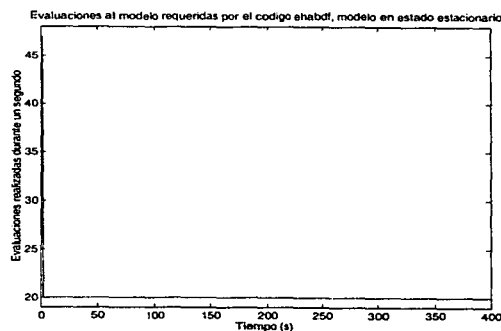
A continuación se presentan el número de matrices calculadas por EHABDF en la aplicación de las pruebas.

Prueba 1. Estado Estacionario		Prueba 2. Estado Transitorio	
Matrices de iteración calculadas	Tiempo de simulación donde se cálculo	Matrices de iteración calculadas	Tiempo de simulación donde se cálculo
1	0.1 seg.	1	0.1 seg.

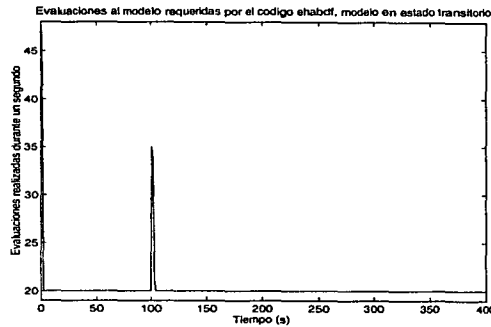
A continuación se presenta un resumen del número de evaluaciones del modelo realizadas por EHABDF en la aplicación de las pruebas.

Prueba	Resumen del número de evaluaciones realizadas al modelo por EHABDF
1	El mayor número de evaluaciones realizadas al modelo por EHABDF fue 29 y se obtuvo al inicio de la simulación debido al cálculo de la matriz de iteración. En cada uno de los pasos de integración posteriores al inicial, EHABDF realizó solo dos evaluaciones del modelo.
2	El mayor número de evaluaciones realizadas al modelo por EHABDF fue 29 y se obtuvo al inicio de la simulación debido al cálculo de la matriz de iteración. En casi todos los pasos de integración posteriores al inicial, se requirieron solo dos evaluaciones del modelo, con excepción de los pasos de integración comprendidos entre los segundos 99 y 101 de la simulación (tiempos en los que transcurre el paso del estado transitorio al nuevo estado estacionario), ya que en estos pasos EHABDF requirió evaluar al modelo entre 2, 3 o 4 veces.

A continuación se presentan las gráficas 3.7.7 y 3.7.8 que muestran el número de evaluaciones realizadas al modelo en cada segundo de simulación alcanzado por EHABDF en la aplicación de las pruebas. Estas gráficas muestran las evaluaciones acumuladas de las evaluaciones realizadas en cada uno de los 10 pasos de integración de 0.1seg. que conforman un segundo de simulación.

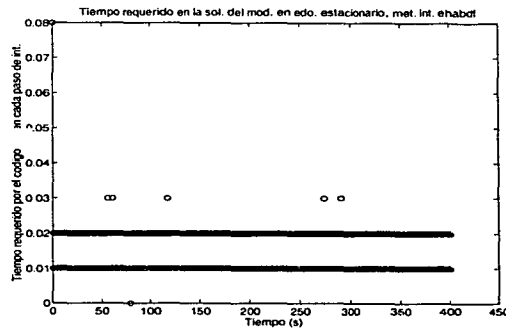


Gráfica 3.7.7 Evaluaciones del modelo realizadas por el código EHABDF en cada segundo, modelo en estado estacionario.

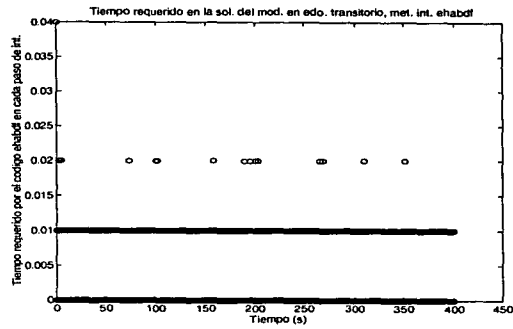


Gráfica 3.7.8 Evaluaciones al modelo realizadas por el código EHABDF en cada segundo, modelo en estado transitorio.

A continuación se presentan las gráficas 3.7.9 y 3.7.10 que muestran el tiempo de UPC requerido para la forma alternativa de solucionar el modelo en cada paso de integración de 0.1 s en la aplicación de las pruebas. El tiempo se midió en segundos con una resolución de hasta centésimas de segundo. En las gráficas se puede observar como el tiempo de CPU requerido en cada paso de integración de 0.1 s es menor a 0.1 s, con ello se cumple que la solución de las pruebas se obtenga en tiempo real.



Gráfica 3.7.9 Tiempo requerido en cada paso de int. de 0.1 s por el código EHABDF, modelo en estado estacionario.



Gráfica 3.7.10 Tiempo requerido en cada paso de int. de 0.1 s por el código EHABDF, modelo en estado transitorio.

En la siguiente tabla se presenta el tiempo total requerido por el código EHABDF en la obtención de la solución del modelo en la aplicación de las pruebas.

Tiempo total requerido (segundos) por el código EHABDF para la obtención de toda la simulación.	
Prueba 1. Modelo En Estado Estacionario	Prueba 2. Modelo En Estado Transitorio
23.88 segundos	24.02 segundos

III.7.3 Memoria requerida por el código EHABDF

La memoria requerida por los argumentos del código EHABDF para poder resolver un sistema de n ecuaciones se presenta en la tabla 3.7.3

Argumento de EHABDF	Tipo de variable	Memoria Requerida (bytes)
H	real*4	4
Hinv	real*4	4
T	real*4	4
Banin	Logical	1
Primpas	Logical	1
Nedas	integer*4	4
psi(n)	real*4	4*n
psipa(n)	real*4	4*n
ypred(n)	real*4	4*n
dypred(n)	real*4	4*n
res(n)	real*4	4*n
tempres(n)	real*4	4*n
matjac(n*n)	real*4	4*(n) ²
del(n)	real*4	4*n
ipvt(n)	integer*4	4*n
Dy(n)	real*4	4*n
y(n)	real*4	4*n
Memoria Total Requerida Por los argumentos de EHABDF	$4*(4 + 10*n + (n)^2) + 2$ bytes	

Tabla 3.7.3

Para la forma alternativa planteada de resolver el modelo, EHABDF debe resolver únicamente las 27 ecuaciones diferenciales del modelo, siguiendo la tabla 3.7.3 se concluye que la memoria requerida en este caso por los argumentos de EHABDF es la presentada en la tabla 3.7.4

Memoria Total Requerida por los Argumentos de EHABDF para la forma alternativa de resolución del modelo.
$4*(4 + 10*27 + (27)^2) + 2 = 4014$ bytes

Tabla 3.7.4

CAPÍTULO IV

CONCLUSIONES

Como se mencionó en el capítulo I, el objetivo de la presente tesis es la validación del código EHABDF para la solución de SEAD's de Modelos de Plantas de Potencia en tiempo real, y su documentación.

La validación del código se realizó en dos etapas; la primera utilizando modelos teóricos sencillos de sistemas de ecuaciones diferenciales ordinarias (SEDO), ver sección II.6 del capítulo II; y la segunda con la aplicación de pruebas a el SEAD de una planta de potencia en tiempo real, ver capítulo III.

Las conclusiones a que se llegaron con el análisis de los resultados obtenidos en las pruebas realizadas se resumen a continuación.

- El código EHABDF puede resolver SEDO's teóricos sencillos, sin dificultades, siempre y cuando estos sistemas sean estables, como lo indican los resultados en II.6.4 y II.6.5; sin embargo, si el modelo es inestable entonces los resultados obtenidos con EHABDF no serán buenos como puede verse en II.6.6.

Para Modelos Dinámicos en Tiempo Real:

- El código EHABDF no podrá obtener la solución del Sistema de Ecuaciones Diferenciales o de un SEAD cuando los valores característicos asociados a dicho sistema, se encuentren fuera de la zona de estabilidad de la fórmula BDF de orden 1.
- El código EHABDF no podrá obtener la solución simultánea de un SEAD cuando la matriz de iteración aproximada por EHABDF sea singular, o casi-singular (mal condicionada), como fue el caso de la resolución simultánea de las 27 ecuaciones diferenciales y las 42 variables algebraicas salidas a la consola del simulador del

modelo de la planta de potencia utilizado para la validación del código, que se presentó en la sección III.5 del capítulo III.

- Para resolver un SEAD de manera óptima (es decir menor tiempo de CPU y mejor aproximación), es necesario que éste sea ordenado en una secuencia de ejecución óptima para evitar problemas de retraso de información que pueda ocasionar una inestabilidad numérica, mayor número de iteraciones y/o requerimiento de un mayor número de variables que necesiten de un valor inicial.
- Los SEAD's de los modelos de plantas de potencia, por lo general, llegan a presentar modificaciones de los valores de las variables de estado y algebraicas fuera de los métodos numéricos, lo cual presenta un problema potencial para que el código EHABDF pueda obtener una buena solución del modelo, ya que EHABDF requiere que el modelo no modifique en forma alguna el valor de las variables que está solucionando. Cuando las variables se modifican a través de ecuaciones limitadoras (ver sección III.1.2.3) con límites constantes, entonces se pueden realizar adaptaciones al modelo y al código EHABDF para que éste pueda trabajar sin problemas al respecto (ver III.6.6). En el caso que las variables se modifiquen con límites variables, este problema requiere de una modelación diferente que no modifique directamente los valores de dichas variables, como se puede ver en la sección III.6.1.2 para el caso de la variable de estado X2GR. Sin embargo, si se hacen modificaciones a las variables que lleven a discontinuidades que no puedan superarse, esto ocasionará un problema potencial al código EHABDF en la obtención de la solución del modelo.
- La obtención de la solución simultánea de un SEAD en tiempo real a través de EHABDF depende en forma cúbica del número de ecuaciones que se resuelven en forma simultánea y de la velocidad en la realización de operaciones aritméticas de la máquina en que se ejecutan.
- En la forma alternativa de solucionar el modelo de la planta de potencia a través de EHABDF, es decir, resolviendo simultáneamente el SEDO del modelo y en forma

secuencial el conjunto de variables algebraicas, la solución del modelo en cada paso de integración de 0.1 se obtuvo en tiempo real, y su aproximación numérica es bastante aceptable para los requerimientos de los simuladores para entrenamiento de plantas de potencia.

IV.1 Mejoras al Código EHABDF

Dentro de las mejoras posibles que se pueden realizar al código EHABDF en trabajos futuros se encuentran las siguientes:

- El código EHABDF puede ser mejorado con la implantación de métodos de cálculo de jacobianos y de solución de sistemas lineales que utilicen matrices ralas.
- El código EHABDF puede ser mejorado con la implantación de herramientas para resolver el problema de matrices mal condicionadas.
- El código EHABDF puede ser mejorado con la implantación de herramientas que resuelvan el problema de las discontinuidades presentes en el modelo.

APÉNDICE A

El presente apéndice presenta la secuencia de ejecución de las ecuaciones, las dependencias de las ecuaciones entre sí y el tipo de ecuación que se trata, del modelo original de la planta de potencia de 400 MW, elegido para la validación del código EHBADF. También presenta los problemas de ordenamiento del modelo original encontrados en la secuencia de ejecución de sus ecuaciones

A.1 Secuencia de Ejecución de las Ecuaciones del Modelo Original

La secuencia de ejecución de las ecuaciones del modelo se presenta en la tabla a.1, la secuencia se indica a través de la columna denominada número de ecuación, la tabla también presenta la variable que se calcula en cada ecuación y las dependencias requeridas en esa ecuación para resolver a la variable en cuestión, también se presenta el tipo de ecuación que se trata original, así como el módulo en que se encuentra la ecuación en el modelo original.

No. Ecu.	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
1	LIMITADOR	EST	X1SC	X1SC	1:CONTSSO	PRIMERA
2	LIMITADOR	EST	X2SC	X2SC	1:CONTSSO	PRIMERA
3	LIMITADOR	ALG	KTSSO	KTSSO WSSO	1:CONTSSO	PRIMERA
4	TRANSDUCTOR	ALG	KCTSSO	KTSSO KCTSSO	1:CONTSSO	PRIMERA
5	TRANSDUCTOR	ALG	CTSSO	CTSSO TSSO	1:CONTSSO	PRIMERA
6	ALGEBRAICA	ALG	C1SC	C1SC KCTSSO CTSSO	1:CONTSSO	PRIMERA
7	ALGEBRAICA	ALG	C2SC	C2SC C1SC	1:CONTSSO	PRIMERA
8	LIMITADOR	ALG	C3SC	C3SC C2SC X1SC	1:CONTSSO	PRIMERA
9	LIMITADOR	EST	X1TR	X1TR	2:CONTRB	PRIMERA
10	LIMITADOR	EST	X2TR	X2TR	2:CONTRB	PRIMERA
11	LIMITADOR	EST	X3TR	X3TR	2:CONTRB	PRIMERA
12	TRANSDUCTOR	ALG	CMWTRO	CMWTRO MWTRO	2:CONTRB	PRIMERA
13	ALGEBRAICA	ALG	C1TR	C1TR CMWTRO	2:CONTRB	PRIMERA
14	ALGEBRAICA	ALG	C2TR	C2TR C1TR X1TR	2:CONTRB	PRIMERA

No. Ecu.	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
15	LIMITADOR	ALG	C3TR	C3TR C2TR	2:CONTRB	PRIMERA
16	TRANSDUCTOR	ALG	CNTR	CNTR	2:CONTRB	PRIMERA
17	ALGEBRAICA	ALG	C4TR	C4TR CNTR	2:CONTRB	PRIMERA
18	ALGEBRAICA	ALG	C5TR	C5TR C4TR	2:CONTRB	PRIMERA
19	ALGEBRAICA	ALG	C6TR	C6TR C5TR X2TR	2:CONTRB	PRIMERA
20	LIMITADOR	EST	X1DM	X1DM	3:CONCOMB	PRIMERA
21	LIMITADOR	EST	X1AR	X1AR	3:CONCOMB	PRIMERA
22	LIMITADOR	EST	X2AR	X2AR	3:CONCOMB	PRIMERA
23	LIMITADOR	EST	X1CB	X1CB	3:CONCOMB	PRIMERA
24	LIMITADOR	EST	X2CB	X2CB	3:CONCOMB	PRIMERA
25	LIMITADOR	ALG	KPSSO	KPSSO	3:CONCOMB	PRIMERA
26	TRANSDUCTOR	ALG	KCPSSO	KPSSO KCPSSO	3:CONCOMB	PRIMERA
27	TRANSDUCTOR	ALG	CPSSO	CPSSO PSSO	3:CONCOMB	PRIMERA
28	ALGEBRAICA	ALG	C1DM	C1DM KCPSSO CPSSO	3:CONCOMB	PRIMERA
29	ALGEBRAICA	ALG	C2DM	C2DM C1DM	3:CONCOMB	PRIMERA
30	LIMITADOR	ALG	C3DM	C3DM C2DM X1DM	3:CONCOMB	PRIMERA
31	ALGEBRAICA	ALG	C4DM	C4DM CNTR	3:CONCOMB	PRIMERA
32	LIMITADOR	ALG	C5DM	C5DM C3DM C4DM	3:CONCOMB	PRIMERA
33	ALGEBRAICA	ALG	C1AR	C1AR C5DM	3:CONCOMB	PRIMERA
34	ALGEBRAICA	ALG	C2AR	C2AR C1AR X2AR	3:CONCOMB	PRIMERA
35	ALGEBRAICA	ALG	C3AR	C3AR C2AR	3:CONCOMB	PRIMERA
36	LIMITADOR	ALG	C4AR	C4AR C3AR X1AR	3:CONCOMB	PRIMERA
37	ALGEBRAICA	ALG	C1CB	C1CB C5DM X2AR	3:CONCOMB	PRIMERA
38	ALGEBRAICA	ALG	C2CB	C2CB C1CB X2CB	3:CONCOMB	PRIMERA
39	ALGEBRAICA	ALG	C3CB	C3CB C2CB	3:CONCOMB	PRIMERA
40	LIMITADOR	ALG	C4CB	C4CB C3CB X1CB	3:CONCOMB	PRIMERA
41	ALGEBRAICA	ALG	KCGR	KCGR XGG	4:CONTRHO	PRIMERA
42	ALGEBRAICA	ALG	C1GR	C1GR X2RC	4:CONTRHO	PRIMERA
43	LIMITADOR	EST	X1GR	X1GR	4:CONTRHO	PRIMERA
44	LIMITADOR	EST	X2GR	X2GR KCWGR	4:CONTRHO	PRIMERA
45	LIMITADOR	EST	X1RC	X1RC	4:CONTRHO	PRIMERA
46	LIMITADOR	EST	X2RC	X2RC	4:CONTRHO	PRIMERA
47	LIMITADOR	EST	X3RC	X3RC	4:CONTRHO	PRIMERA
48	LIMITADOR	ALG	KTRH	KTRH WSSO	4:CONTRHO	PRIMERA
49	TRANSDUCTOR	ALG	KTRH	KTRH KCTRH	4:CONTRHO	PRIMERA
50	TRANSDUCTOR	ALG	CTRHO	CTRHO TRHO	4:CONTRHO	PRIMERA

No. Ecu.	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
51	ALGEBRAICA	ALG	C1RC	C1RC KCTRH CTRHO	4:CONTRHO	PRIMERA
52	ALGEBRAICA	ALG	C2RC	C2RC C1RC	4:CONTRHO	PRIMERA
53	LIMITADOR	ALG	C3RC	C3RC C2RC X1RC	4:CONTRHO	PRIMERA
54	ALGEBRAICA	ALG	WG	WG WAR WGR WFL	5:HOGAR	PRIMERA
55	ALGEBRAICA	ALG	AUX	AUX WAR WGR WFL	5:HOGAR	PRIMERA
56	ALGEBRAICA	ALG	TFNI	TFNI AUX WG	5:HOGAR	PRIMERA
57	ALGEBRAICA	ALG	UXGG	UXGG XGG	6:GSHTLSRAD	PRIMERA
58	ALGEBRAICA	ALG	KUWWGM	KUWWGM WFL	6:GSHTLSRAD	PRIMERA
59	ALGEBRAICA	ALG	UWWGM	UWWGM UXGG KUWWGM	6:GSHTLSRAD	PRIMERA
60	ALGEBRAICA	ALG	ZFNI	ZFNI WG UWWGM	6:GSHTLSRAD	PRIMERA
61	ALGEBRAICA	ALG	TWWGE	TWWGE WG TFNI UWWGM ZFNI	6:GSHTLSRAD	PRIMERA
62	ALGEBRAICA	ALG	QWWGM	QWWGM UWWGM TWWGE	6:GSHTLSRAD	PRIMERA
63	ALGEBRAICA	ALG	TWWGO	TWWGO TWWGE QWWGM WG	6:GSHTLSRAD	PRIMERA
64	ALGEBRAICA	ALG	QGP	QGP WG TWWGO	7:GAS	PRIMERA
65	ALGEBRAICA	ALG	ZWWG	ZWWG WG WGR	7:GAS	PRIMERA
66	ALGEBRAICA	ALG	ZGRH	ZGRH ZWWG	7:GAS	PRIMERA
67	ALGEBRAICA	ALG	ZGEC	ZGEC ZWWG	7:GAS	PRIMERA
68	ALGEBRAICA	ALG	QRH	QRH ZGRH QGP	7:GAS	PRIMERA
69	ALGEBRAICA	ALG	QEC	QEC ZGEC QGP	7:GAS	PRIMERA
70	ALGEBRAICA	ALG	QSH	QSH QGP QRH QEC	7:GAS	PRIMERA
71	ALGEBRAICA	ALG	RDRW	RDRW	9:DOMO	PRIMERA
72	ALGEBRAICA	ALG	TDRS	TDRS	9:DOMO	PRIMERA
73	ALGEBRAICA	ALG	HDRW	HDRW	9:DOMO	PRIMERA
74	ALGEBRAICA	ALG	HDRS	HDRS	9:DOMO	PRIMERA
75	ALGEBRAICA	ALG	PDRS	PDRS	9:DOMO	PRIMERA
76	ALGEBRAICA	ALG	QWWMW	QWWMW TDRS	10:PRDH2O	PRIMERA
77	ALGEBRAICA	ALG	PSSO	PSSO	11:SPRTHTR	PRIMERA
78	ALGEBRAICA	ALG	WDRS	WDRS PDRS PSSO	11:SPRTHTR	PRIMERA
79	ALGEBRAICA	ALG	TSSO	TSSO	11:SPRTHTR	PRIMERA
80	ALGEBRAICA	ALG	SSSO	SSSO	11:SPRTHTR	PRIMERA
81	ALGEBRAICA	ALG	WSSO	WSSO ACV PSSO	12:STMFLW	PRIMERA
82	ALGEBRAICA	ALG	WHP	WHP WSSO	12:STMFLW	PRIMERA
83	ALGEBRAICA	ALG	ZDR(1)	ZDR(1)	13:DRMWTRFL	PRIMERA
84	ALGEBRAICA	ALG	ZDR(2)	ZDR(2)	13:DRMWTRFL	PRIMERA

No. Ecu.	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
85	ALGEBRAICA	ALG	ZDR(3)	ZDR(3)	13:DRMWTRFL	PRIMERA
86	ALGEBRAICA	ALG	ZDR(4)	ZDR(4) ZDR(3)	13:DRMWTRFL	PRIMERA
87	ALGEBRAICA	ALG	ZDR(5)	ZDR(5) HDRS ZDR(1) HDRW ZDR(3) RDRW ZDR(2)	13:DRMWTRFL	PRIMERA
88	ALGEBRAICA	ALG	ZDR(6)	ZDR(6) QWWMW WDRS HDRS	13:DRMWTRFL	PRIMERA
89	ALGEBRAICA	ALG	ZDR(7)	ZDR(7) WDRS ZDR(5) ZDR(4) ZDR(6)	13:DRMWTRFL	PRIMERA
90	ALGEBRAICA	ALG	ZDR(8)	ZDR(8) ZDR(5) ZDR(4)	13:DRMWTRFL	PRIMERA
91	ALGEBRAICA	ALG	ZDR(9)	ZDR(9) ZDR(7) ZDR(8)	13:DRMWTRFL	PRIMERA
92	ALGEBRAICA	ALG	WFW	WFW ZDR(9)	13:DRMWTRFL	PRIMERA
93	ALGEBRAICA	ALG	PRHO	PRHO	14:RHTR	PRIMERA
94	ALGEBRAICA	ALG	TRHO	TRHO	14:RHTR	PRIMERA
95	ALGEBRAICA	ALG	PHPO	PHPO PRHO	14:RHTR	PRIMERA
96	ALGEBRAICA	ALG	WHPO	WHPO WHP	15:HPTRB	PRIMERA
97	ALGEBRAICA	ALG	NISEN	NISEN WHP	15:HPTRB	PRIMERA
98	ALGEBRAICA	ALG	HHPOI	HHPOI SSSO PHPO	15:HPTRB	PRIMERA
99	ALGEBRAICA	ALG	HHPO	HHPO HHPOI NISEN	15:HPTRB	PRIMERA
100	ALGEBRAICA	ALG	MWHP	MWHP WHP HHPO	15:HPTRB	PRIMERA
101	ALGEBRAICA	ALG	THPO	THPO PHPO HHPO	15:HPTRB	PRIMERA
102	ALGEBRAICA	ALG	HLPO	HLPO	16:CNDNSR	PRIMERA
103	ALGEBRAICA	ALG	WRHO	WRHO PRHO	17:LPTRB	PRIMERA
104	ALGEBRAICA	ALG	WLP	WLP WRHO	17:LPTRB	PRIMERA
105	ALGEBRAICA	ALG	MWLP	MWLP WLP HLPO	17:LPTRB	PRIMERA
106	ALGEBRAICA	ALG	MWGNPU	MWGNPU	18:TRBGNRTR	PRIMERA
107	ALGEBRAICA	ALG	MWGN	MWGN MWGNPU	18:TRBGNRTR	PRIMERA
108	ALGEBRAICA	ALG	MWTRO	MWTRO MWHP MWLP	18:TRBGNRTR	PRIMERA
109	ALGEBRAICA	ALG	HCPO	HCPO	19:CNDPMP	PRIMERA
110	ALGEBRAICA	ALG	HLHD	HLHD WSSO	20:LPHDTR	PRIMERA
111	ALGEBRAICA	ALG	HDED	HDED WSSO	21:DRDR	PRIMERA
112	ALGEBRAICA	ALG	HHHD	HHHD WSSO	22:HPHTR	PRIMERA
113	ALGEBRAICA	ALG	HFPO	HFPO	23:BLRFDPMP	PRIMERA
114	ALGEBRAICA	ALG	Y2SC	Y2SC X2SC	1:CONTSSO	SEGUNDA
115	TRANSDUCTOR	ALG	WSY	WSY Y2SC	1:CONTSSO	SEGUNDA
116	DERIVADA	DER	DX1SC	DX1SC C2SC	1:CONTSSO	SEGUNDA
117	DERIVADA	DER	DX2SC	DX2SC C3SC X2SC	1:CONTSSO	SEGUNDA

No. Ecu.	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
118	TRANSDUCTOR	ALG	ACV	ACV X3TR	2:CONTRB	SEGUNDA
119	DERIVADA	DER	DX1TR	DX1TR C1TR	2:CONTRB	SEGUNDA
120	DERIVADA	DER	DX2TR	DX2TR C3TR X2TR	2:CONTRB	SEGUNDA
121	DERIVADA	DER	DX3TR	DX3TR C6TR X3TR	2:CONTRB	SEGUNDA
122	TRANSDUCTOR	ALG	WAR	WAR X2AR	3:CONCOMB	SEGUNDA
123	TRANSDUCTOR	ALG	WFL	WFL X2CB	3:CONCOMB	SEGUNDA
124	DERIVADA	DER	DX1DM	DX1DM C2DM	3:CONCOMB	SEGUNDA
125	DERIVADA	DER	DX1AR	DX1AR C3AR	3:CONCOMB	SEGUNDA
126	DERIVADA	DER	DX2AR	DX2AR C4AR X2AR	3:CONCOMB	SEGUNDA
127	DERIVADA	DER	DX1CB	DX1CB C3CB	3:CONCOMB	SEGUNDA
128	DERIVADA	DER	DX2CB	DX2CB C4CB X2CB	3:CONCOMB	SEGUNDA
129	ALGEBRAICA	ALG	KWGR	KWGR WAR	4:CONTRHO	SEGUNDA
130	TRANSDUCTOR	ALG	KCWGR	KCWGR KWGR	4:CONTRHO	SEGUNDA
131	TRANSDUCTOR	ALG	WGR	WGR X2GR	4:CONTRHO	SEGUNDA
132	DERIVADA	DER	DX1GR	DX1GR C1GR KCGR	4:CONTRHO	SEGUNDA
133	DERIVADA	DER	DX2GR	DX2GR X1GR X2GR	4:CONTRHO	SEGUNDA
134	ALGEBRAICA	ALG	Y3RC	Y3RC X3RC	4:CONTRHO	SEGUNDA
135	TRANSDUCTOR	ALG	WRY	WRY Y3RC	4:CONTRHO	SEGUNDA
136	TRANSDUCTOR	ALG	XGG	XGG X2RC	4:CONTRHO	SEGUNDA
137	DERIVADA	DER	DX1RC	DX1RC C2RC	4:CONTRHO	SEGUNDA
138	DERIVADA	DER	DX2RC	DX2RC C3RC X2RC	4:CONTRHO	SEGUNDA
139	DERIVADA	DER	DX3RC	DX3RC C3RC X3RC	4:CONTRHO	SEGUNDA
140	DERIVADA	DER	DHECO	DHECO WFW QEC	8:ECNMDR	SEGUNDA
141	ALGEBRAICA	ALG	ZDO(1)	ZDO(1)	9:DOMO	SEGUNDA
142	ALGEBRAICA	ALG	ZDO(2)	ZDO(2)	9:DOMO	SEGUNDA
143	ALGEBRAICA	ALG	ZDO(3)	ZDO(3)	9:DOMO	SEGUNDA
144	ALGEBRAICA	ALG	ZDO(4)	ZDO(4) ZDO(3)	9:DOMO	SEGUNDA
145	ALGEBRAICA	ALG	ZDO(5)	ZDO(5) HDRS ZDO(1) HDRW ZDO(3) RDRW ZDO(2)	9:DOMO	SEGUNDA
146	ALGEBRAICA	ALG	ZDO(6)	ZDO(6) QWWMW WDRS HDRS	9:DOMO	SEGUNDA
147	ALGEBRAICA	ALG	ZDO(7)	ZDO(7) WDRS ZDO(5) ZDO(4) ZDO(6)	9:DOMO	SEGUNDA
148	ALGEBRAICA	ALG	ZDO(8)	ZDO(8) ZDO(5) ZDO(4)	9:DOMO	SEGUNDA
149	ALGEBRAICA	ALG	ZDO(9)	ZDO(9) ZDO(7) ZDO(8)	9:DOMO	SEGUNDA

No. Ecu.	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
150	ALGEBRAICA	ALG	ZDO(10)	ZDO(10) ZDO(9) WDRS ZDO(4)	9:DOMO	SEGUNDA
151	DERIVADA	DER	DRDRS	DRDRS ZDO(10)	9:DOMO	SEGUNDA
152	DERIVADA	DER	DTWWM	DTWWM QWWGM QWWMW	10:PRDH2O	SEGUNDA
153	DERIVADA	DER	DRSSO	DRSSO WDRS WSY WSSO	11:SPRTHTR	SEGUNDA
154	DERIVADA	DER	DHSSO	DHSSO WDRS HDRS WSY HFPO WSSO QSH	11:SPRTHTR	SEGUNDA
155	DERIVADA	DER	DRRHO	DRRHO WHPO WRY WRHO	14:RHTR	SEGUNDA
156	DERIVADA	DER	DHRHO	DHRHO WHPO HHPO WRY HFPO WRHO QRH	14:RHTR	SEGUNDA
157	DERIVADA	DER	DNTR	DNTR MWTRO MWGN	18:TRBGNRTR	SEGUNDA
158	DERIVADA	DER	DDELTA	DDELTA	18:TRBGNRTR	SEGUNDA
159	DERIVADA	DER	DHIHO	DHIHO HCPO HLHD	20:LPHDHTR	SEGUNDA
160	DERIVADA	DER	DHDEW	DHDEW HDED	21:DRDR	SEGUNDA
161	DERIVADA	DER	DHHHO	DHHHO HFPO HHHH	22:HPHTR	SEGUNDA

Tabla a.1 Secuencia de ejecución de las ecuaciones del modelo original

Como se puede apreciar en la tabla a.1 el modelo consta de 161 ecuaciones algebraico-diferenciales, 27 de ellas son ecuaciones diferenciales que resuelven a las 27 variables de estado del modelo y el resto son ecuaciones algebraicas. De las ecuaciones algebraicas 42 de ellas resuelven a 42 variables algebraicas que son salidas a los tableros del simulador, el resto de las variables algebraicas calculadas solo son variables auxiliares que se utilizan para la evaluación de las ecuaciones diferenciales o para la resolución de las 42 variables algebraicas ya mencionadas. A continuación se presenta en la tabla a.2 las 27 derivadas del modelo y en la tabla a.3 las 42 variables algebraicas.

No. ec	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Variable de estado correspondiente	Módulo	Sección del módulo donde se calcula
116	DERIVADA	DER	DX1SC	X1SC	1:CONTSSO	2ª
117	DERIVADA	DER	DX2SC	X2SC	1:CONTSSO	2ª
119	DERIVADA	DER	DX1TR	X1TR	2:CONTRB	2ª
120	DERIVADA	DER	DX2TR	X2TR	2:CONTRB	2ª
121	DERIVADA	DER	DX3TR	X3TR	2:CONTRB	2ª
124	DERIVADA	DER	DX1DM	X1DM	3:CONCOMB	2ª
125	DERIVADA	DER	DX1AR	X1AR	3:CONCOMB	2ª
126	DERIVADA	DER	DX2AR	X2AR	3:CONCOMB	2ª
127	DERIVADA	DER	DX1CB	X1CB	3:CONCOMB	2ª
128	DERIVADA	DER	DX2CB	X2CB	3:CONCOMB	2ª
132	DERIVADA	DER	DX1GR	X1GR	4:CONTRHO	2ª
133	DERIVADA	DER	DX2GR	X2GR	4:CONTRHO	2ª
137	DERIVADA	DER	DX1RC	X1RC	4:CONTRHO	2ª
138	DERIVADA	DER	DX2RC	X2RC	4:CONTRHO	2ª
139	DERIVADA	DER	DX3RC	X3RC	4:CONTRHO	2ª
140	DERIVADA	DER	DHECO	HECO	8:ECNMDR	2ª
151	DERIVADA	DER	DRDRS	RDRS	9:DOMO	2ª
152	DERIVADA	DER	DTWWM	TWWM	10:PRDH2O	2ª
153	DERIVADA	DER	DRSSO	DRSSO	11:SPRTHTR	2ª
154	DERIVADA	DER	DHSSO	HSSO	11:SPRTHTR	2ª
155	DERIVADA	DER	DRRHO	RRHO	14:RHTR	2ª
156	DERIVADA	DER	DHRHO	HRHO	14:RHTR	2ª
157	DERIVADA	DER	DNTR	NTR	18:TRBGNRT R	2ª
158	DERIVADA	DER	DDELTA	DELTA	18:TRBGNRT R	2ª
159	DERIVADA	DER	DH1HO	H1HO	20:LPHDHTR	2ª
160	DERIVADA	DER	DHDEW	HDEW	21:DRDR	2ª
161	DERIVADA	DER	DHHHO	HHHO	22:HPHTR	2ª

Tabla a.2 Derivadas y variables de estado del modelo

VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA	VAR. ALGEBRAICA
WG	HDRW	WFW	WFL	TRHO	HFPO
TFN1	RDRW	WSSO	XGG	WHPO	HCPO
TWWGO	WDRS	WSY	QSH	HHPO	MWGN
QWWGM	PSSO	ACV	QRH	THPO	HLHD
HDRS	SSSO	WRY	QEC	MWHP	HDED
TDRS	TSSO	WAR	PHPO	WRHO	HHHD
PDRS	QWWMW	WGR	PRHO	MWLP	MWTRO

Tabla a.3 Variables algebraicas de salida a los tableros del simulador

Dentro de las ecuaciones del modelo se encuentra un tipo especial de ecuación llamada limitador, ésta verifica que el valor de una variable obtenido a través del método integrador u otro, se encuentre dentro de rangos permisibles de operación, por lo que propiamente no es una ecuación, sino una serie de instrucciones que vigilan que el valor esté dentro de ciertos límites. A continuación se presentan en la tabla a.4 las variables de estado del modelo que están siendo limitadas.

Módulo	No. ec	Tipo de ecuación	Tipo Var que modif	Variable Modificada	Rango permisible
1:CONTSSO	1	LIMITADOR	EST	X1SC	1 <= valor <= 5
1:CONTSSO	2	LIMITADOR	EST	X2SC	1 <= valor <= 5
2:CONTRB	9	LIMITADOR	EST	X1TR	-1 <= valor <= 1
2:CONTRB	10	LIMITADOR	EST	X2TR	0 <= valor <= 5
2:CONTRB	11	LIMITADOR	EST	X3TR	1 <= valor <= 5
3:CONCOMB	20	LIMITADOR	EST	X1DM	1 <= valor <= 5
3:CONCOMB	21	LIMITADOR	EST	X1AR	1 <= valor <= 5
3:CONCOMB	22	LIMITADOR	EST	X2AR	1 <= valor <= 5
3:CONCOMB	23	LIMITADOR	EST	X1CB	1 <= valor <= 5
3:CONCOMB	24	LIMITADOR	EST	X2CB	1 <= valor <= 5
4:CONTRHO	43	LIMITADOR	EST	X1GR	1 <= valor <= 5
4:CONTRHO	44	LIMITADOR	EST	X2GR	1 <= valor <= 5
4:CONTRHO	45	LIMITADOR	EST	X1RC	1 <= valor <= 5
4:CONTRHO	46	LIMITADOR	EST	X2RC	1 <= valor <= 5
4:CONTRHO	47	LIMITADOR	EST	X3RC	1 <= valor <= 5

Tabla a.4 Variables de estado limitadas

Las 27 variables de estado y las 42 variables algebraicas que son salidas a los tableros del simulador son las variables que entran a los diferentes módulos del modelo, en la tabla a.5 se presentan estas entradas a los diferentes módulos del modelo.

Módulo	Entrada	No. módulo de donde proviene	Nombre de módulo de donde proviene	Tipo de variable	Modificada mediante	tiene valor de inicio?
1:CONTSSO	TSSO	11	SPRTHTR	ALG.	ECUACIÓN	NO
	WSSO	12	STMFLW	ALG.	ECUACIÓN	NO
2:CONTRB	MWTRO	18	TRBGNRTR	ALG.	ECUACIÓN	NO
3:CONCOMB	PSSO	11	SPRTHTR	ALG.	ECUACIÓN	NO
4:CONTRHO	WAR	3	CONCOMB	DER.	TRANSDUCE	SI
	TRHO	14	RHTR	ALG.	ECUACIÓN	NO
	WSSO	12	STMFLW	ALG.	ECUACIÓN	NO
5:HOGAR	WAR	3	CONCOMB	DER.	TRANSDUCE	SI
	WR	4	CONTRHO	DER.	TRANSDUCE	SI
	WFL	3	CONCOMB	DER.	TRANSDUCE	SI
6:GSHTLSRAD	WFL	3	CONCOMB	DER.	TRANSDUCE	SI
	XGG	4	CONTRHO	DER.	TRANSDUCE	SI
	WG	5	HOGAR	ALG.	ECUACIÓN	NO
	TFN1	5	HOGAR	ALG.	ECUACIÓN	NO
7:GAS	WGR	4	CONTRHO	DER.	TRANSDUCE	SI
	WG	5	HOGAR	ALG.	ECUACIÓN	NO
	TWWGO	6	GSHTLSRAD	ALG.	ECUACIÓN	NO
8:ECNMDR	QEC	7	GAS	ALG.	ECUACIÓN	NO
	WFW	13	DRMWTRFL	ALG.	ECUACIÓN	NO
9:DOMO	WDRS	11	SPRTHTR	ALG.	ECUACIÓN	NO
	QWWMW	10	PRDH2O	ALG.	ECUACIÓN	NO
10:PRDH2O	TDRS	9	DOMO	ALG.	ECUACIÓN	SI
	QWWGM	6	GSHTLSRAD	ALG.	ECUACIÓN	NO
11:SPRTHTR	WSY	1	CONTSSO	DER.	TRANSDUCE	SI
	HDRS	9	DOMO	ALG.	ECUACIÓN	NO
	PDRS	9	DOMO	ALG.	ECUACIÓN	NO
	WSSO	12	STMFLW	ALG.	ECUACIÓN	NO
	QSH	7	GAS	ALG.	ECUACIÓN	NO
	HFPO	23	BLRFDMP	ALG.	ECUACIÓN	NO

Módulo	Entrada	No. módulo de donde proviene	Nombre de módulo de donde proviene	Tipo de variable	Modificada mediante	tiene valor de inicio?
12:STMFLW	ACV	2	CONTRB	DER.	TRANSDUCE	SI
	PSSO	11	SPRTHTR	ALG.	ECUACIÓN	NO
13:DRMWTRFL	HDRS	9	DOMO	ALG.	ECUACIÓN	NO
	HDRW	9	DOMO	ALG.	ECUACIÓN	NO
	RDRW	9	DOMO	ALG.	ECUACIÓN	NO
	WDRS	11	SPRTHTR	ALG.	ECUACIÓN	NO
	QWWMW	10	PRDH2O	ALG.	ECUACIÓN	NO
14:RHTR	WRY	4	CONTRHO	DER.	TRANSDUCE	SI
	QRH	7	GAS	ALG.	ECUACIÓN	NO
	HFPO	23	BLRFDMP	ALG.	ECUACIÓN	NO
	WHPO	15	HPTRB	ALG.	ECUACIÓN	NO
	HHPO	15	HPTRB	ALG.	ECUACIÓN	NO
	WRHO	17	LPTRB	ALG.	ECUACIÓN	NO
15:HPTRB	PHPO	14	RHTR	ALG.	ECUACIÓN	SI
	SSSO	11	SPRTHTR	ALG.	ECUACIÓN	NO
	WHP	12	STMFLW	ALG.	ECUACIÓN	NO
17:LPTRB	PRHO	14	RHTR	ALG.	ECUACIÓN	NO
	HLPO	16	CNDNSR	ALG.	ECUACIÓN	NO
18:TRBGNRTR	MWHP	15	HPTRB	ALG.	ECUACIÓN	NO
	MWLP	17	LPTRB	ALG.	ECUACIÓN	NO
20:LPFDHTR	WSSO	12	STMFLW	ALG.	ECUACIÓN	NO
	HCPO	19	CNDPMP	ALG.	ECUACIÓN	NO
21:DRDR	WSSO	12	STMFLW	ALG.	ECUACIÓN	NO
22:HPHTR	WSSO	12	STMFLW	ALG.	ECUACIÓN	NO
	HFPO	23	BLRFDMP	ALG.	ECUACIÓN	NO

Tabla a.5 Entradas a los módulos del modelo

A.2 Ordenamiento del Modelo Original

Con la información presentada en la tabla a.1 de las dependencias de las ecuaciones junto con su secuencia de ejecución, se formó la matriz estructural del modelo [Ojeda_2], esta matriz contiene en forma binaria la estructura de las dependencias de las ecuaciones entre sí, de la siguiente forma:

$$c_{i,j} = \begin{cases} 1, & \text{si la ecuación } i \text{ depende de la ecuación } j \\ 0, & \text{en caso contrario} \end{cases}$$

El orden en que se ejecutan las ecuaciones dentro del modelo es el orden en que aparecen en los renglones de la matriz.

La matriz estructural del modelo permite conocer que ecuaciones se encuentran dentro de ciclos iterativos o que ecuaciones se encuentran en un orden de ejecución inadecuado. En el apéndice B se presenta la matriz estructural obtenida del modelo. Ésta mostró la existencia de ecuaciones que tienen un orden de ejecución inadecuado ya que dependen de variables que no tienen un valor inicial y que son calculadas posteriormente; esto hace que las ecuaciones sean evaluadas incorrectamente al inicio de la simulación y que en el transcurso de la simulación sean evaluadas con información atrasada. A continuación se presentan las ecuaciones encontradas.

# ec.	Tipo de ecuación	Módulo donde se ubica	Variable Modificada	Tipo Variable	Dependencia calculada posteriormente y sin valor de inicio	No. ecuación donde se calcula la dependencia
3	LIMITADOR	1:CONTSSO	KTSSO	ALG	WSSO	81
5	TRANSDUCTOR	1:CONTSSO	CTSSO	ALG	TSSO	71
12	TRANSDUCTOR	2:CONTRB	CMWTRO	ALG	MWTRO	108
27	TRANSDUCTOR	3:CONCOMB	CPSSO	ALG	PSSO	77
50	TRANSDUCTOR	4:CONTRHO	CTRHO	ALG	TRHO	94

Tabla a.6 Ecuaciones con un orden de ejecución incorrecto

APÉNDICE C

Se presenta el nuevo ordenamiento realizado a la secuencia de ejecución de las ecuaciones del modelo original.

C.1 Ordenamiento de las Ecuaciones del Modelo

Los problemas de ordenamiento de la secuencia de ejecución de las variables del modelo, plantean un problema potencial para la resolución del modelo a través del código EHABDF, debido a que gran parte de las ecuaciones algebraicas solo son cálculos auxiliares que se requieren para la evaluación de las derivadas o de las 42 variables algebraicas que son salidas a los tableros del simulador. Por ello, éstas variables auxiliares no deben resolverse en forma simultánea, sin embargo, deben tener una secuencia de ejecución correcta, para evitar retrasos de información en la resolución del modelo. Debido a esto, se decidió ordenar la secuencia de ejecución de las ecuaciones del modelo.

Para realizar el ordenamiento de las ecuaciones del modelo, se aplicó un algoritmo de ordenamiento denominado Algoritmo de Ledet [Ojeda_2]. Este algoritmo trabaja a partir de la matriz estructural inicial obtenida del modelo y en base a algoritmos de Teoría de Gráficas busca minimizar los ciclos iterativos de las ecuaciones presentes en el modelo. Proporcionando como resultado una Matriz Estructural Final la cual contiene el nuevo ordenamiento de las ecuaciones del modelo.

La matriz estructural final reportada por el algoritmo de Ledet (ver apéndice D), muestra el ordenamiento de las ecuaciones con el menor número posible de ciclos iterativos entre ellas.

En el caso del modelo la matriz estructural final obtenida, resulto libre de ciclos, pues se obtuvo una matriz triangular inferior.

El nuevo ordenamiento obtenido por Ledet requería de varios cambios de posición de las ecuaciones del modelo, así como el traslado de ecuaciones de un módulo a otro, sin embargo, al analizar la estructura de la matriz inicial de dependencias de las ecuaciones del modelo (ver apéndice B), se puede ver que esta matriz es muy parecida a una matriz triangular inferior, salvo por algunas ecuaciones que hacen que la estructura de la matriz no sea triangular inferior, ya que éstas dependen de variables calculadas posteriormente y que no tienen un valor inicial. Las dependencias de dichas ecuaciones fueron analizadas para ver si podían ser evaluadas previamente a la ecuación en donde eran requeridas y obtener así una matriz triangular inferior sin necesidad de realizar grandes cambios de posición a las ecuaciones y ejecución de módulos del modelo. Los resultados de los análisis indicaron que esto era posible, por lo que sólo se tuvo necesidad de cambiar el orden local de ejecución de algunas ecuaciones dentro de los 4 primeros módulos del modelo:

1.-Contsso	2.- Contrb	3.- Concomb	4.- Contrho
------------	------------	-------------	-------------

Las ecuaciones cambiaron de posición dentro de los mismos módulos, por lo que al no transferirse ecuaciones de un módulo a otro, no se requirió cambiar la secuencia de ejecución de los módulos. Las entradas y salidas a cada módulo tampoco sufrieron cambios. La tabla c.1 presenta el nuevo ordenamiento obtenido con estos cambios del modelo y la matriz estructural final asociada a este ordenamiento se presenta en el apéndice E. Al modelo ordenado se le denominó modelo ordenado explícito, debido a que sus ecuaciones se encuentran ordenadas y sus expresiones están escritas en forma explícita.

No. Ecu. orden	No. Ecu. Desor.	Tipo de ecuación	Tipo Var.	Variable modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
1	1	LIMITADOR	EST	X1SC	X1SC	1:CONTSSO	PRIMERA
2	2	LIMITADOR	EST	X2SC	X2SC	1:CONTSSO	PRIMERA
3	9	LIMITADOR	EST	X1TR	X1TR	2:CONTRB	PRIMERA
4	10	LIMITADOR	EST	X2TR	X2TR	2:CONTRB	PRIMERA
5	11	LIMITADOR	EST	X3TR	X3TR	2:CONTRB	PRIMERA
6	16	TRANSDUCTOR	ALG	CNTR	CNTR	2:CONTRB	PRIMERA

No. Ecu. orden	No. Ecu. Desor.	Tipo de ecuación	Tipo Var.	Variable modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
7	17	ALGEBRAICA	ALG	C4TR	C4TR CNTR	2:CONTRB	PRIMERA
8	18	ALGEBRAICA	ALG	C5TR	C5TR C4TR	2:CONTRB	PRIMERA
9	19	ALGEBRAICA	ALG	C6TR	C6TR C5TR X2TR	2:CONTRB	PRIMERA
10	118	TRANSDUCTOR	ALG	ACV	ACV X3TR	2:CONTRB	PRIMERA
11	20	LIMITADOR	EST	X1DM	X1DM	3:CONCOMB	PRIMERA
12	21	LIMITADOR	EST	X1AR	X1AR	3:CONCOMB	PRIMERA
13	22	LIMITADOR	EST	X2AR	X2AR	3:CONCOMB	PRIMERA
14	23	LIMITADOR	EST	X1CB	X1CB	3:CONCOMB	PRIMERA
15	24	LIMITADOR	EST	X2CB	X2CB	3:CONCOMB	PRIMERA
16	25	LIMITADOR	ALG	KPSSO	KPSSO	3:CONCOMB	PRIMERA
17	26	TRANSDUCTOR	ALG	KCPSSO	KPSSO KCPSSO	3:CONCOMB	PRIMERA
18	122	TRANSDUCTOR	ALG	WAR	WAR X2AR	3:CONCOMB	PRIMERA
19	123	TRANSDUCTOR	ALG	WFL	WFL X2CB	3:CONCOMB	PRIMERA
20	43	LIMITADOR	EST	X1GR	X1GR	4:CONTRHO	PRIMERA
21	129	ALGEBRAICA	ALG	KWGR	KWGR WAR	4:CONTRHO	PRIMERA
22	130	TRANSDUCTOR	ALG	KCWGR	KCWGR KWGR	4:CONTRHO	PRIMERA
23	44	LIMITADOR	EST	X2GR	X2GR KCWGR	4:CONTRHO	PRIMERA
24	131	TRANSDUCTOR	ALG	WGR	WGR X2GR	4:CONTRHO	PRIMERA
25	45	LIMITADOR	EST	X1RC	X1RC	4:CONTRHO	PRIMERA
26	46	LIMITADOR	EST	X2RC	X2RC	4:CONTRHO	PRIMERA
27	47	LIMITADOR	EST	X3RC	X3RC	4:CONTRHO	PRIMERA
28	136	TRANSDUCTOR	ALG	XGG	XGG X2RC	4:CONTRHO	PRIMERA
29	54	ALGEBRAICA	ALG	WG	WG WAR WGR WFL	5:HOGAR	PRIMERA
30	55	ALGEBRAICA	ALG	AUX	AUX WAR WGR WFL	5:HOGAR	PRIMERA
31	56	ALGEBRAICA	ALG	TFN1	TFN1 AUX WG	5:HOGAR	PRIMERA
32	57	ALGEBRAICA	ALG	UXGG	UXGG XGG	6:GSHTLSRAD	PRIMERA
33	58	ALGEBRAICA	ALG	KUWWGM	KUWWGM WFL	6:GSHTLSRAD	PRIMERA
34	59	ALGEBRAICA	ALG	UWWGM	UWWGM UXGG KUWWGM	6:GSHTLSRAD	PRIMERA
35	60	ALGEBRAICA	ALG	ZFN1	ZFN1 WG UWWGM	6:GSHTLSRAD	PRIMERA
36	61	ALGEBRAICA	ALG	TWWGE	TWWGE WG TFN1 UWWGM ZFN1	6:GSHTLSRAD	PRIMERA
37	62	ALGEBRAICA	ALG	QWWGM	QWWGM UWWGM TWWGE	6:GSHTLSRAD	PRIMERA
38	63	ALGEBRAICA	ALG	TWWGO	TWWGO TWWGE QWWGM WG	6:GSHTLSRAD	PRIMERA

No. Ecu. orden	No. Ecu. Desor.	Tipo de ecuación	Tipo Var.	Variable modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
39	64	ALGEBRAICA	ALG	QGP	QGP WG TWWGO	7:GAS	PRIMERA
40	65	ALGEBRAICA	ALG	ZWWG	ZWWG WG WGR	7:GAS	PRIMERA
41	66	ALGEBRAICA	ALG	ZGRH	ZGRH ZWWG	7:GAS	PRIMERA
42	67	ALGEBRAICA	ALG	ZGEC	ZGEC ZWWG	7:GAS	PRIMERA
43	68	ALGEBRAICA	ALG	QRH	QRH ZGRH QGP	7:GAS	PRIMERA
44	69	ALGEBRAICA	ALG	QEC	QEC ZGEC QGP	7:GAS	PRIMERA
45	70	ALGEBRAICA	ALG	QSH	QSH QGP QRH QEC	7:GAS	PRIMERA
46	71	ALGEBRAICA	ALG	RDRW	RDRW	9:DOMO	PRIMERA
47	72	ALGEBRAICA	ALG	TDRS	TDRS	9:DOMO	PRIMERA
48	73	ALGEBRAICA	ALG	HDRW	HDRW	9:DOMO	PRIMERA
49	74	ALGEBRAICA	ALG	HDRS	HDRS	9:DOMO	PRIMERA
50	75	ALGEBRAICA	ALG	PDRS	PDRS	9:DOMO	PRIMERA
51	76	ALGEBRAICA	ALG	QWWMW	QWWMW TDRS	10:PRDH2O	PRIMERA
52	77	ALGEBRAICA	ALG	PSSO	PSSO	11:SPRTHTR	PRIMERA
53	78	ALGEBRAICA	ALG	WDRS	WDRS PDRS PSSO	11:SPRTHTR	PRIMERA
54	79	ALGEBRAICA	ALG	TSSO	TSSO	11:SPRTHTR	PRIMERA
55	80	ALGEBRAICA	ALG	SSSO	SSSO	11:SPRTHTR	PRIMERA
56	81	ALGEBRAICA	ALG	WSSO	WSSO ACV PSSO	12:STMFLW	PRIMERA
57	82	ALGEBRAICA	ALG	WHP	WHP WSSO	12:STMFLW	PRIMERA
58	83	ALGEBRAICA	ALG	ZDR(1)	ZDR(1)	13:DRMWTRFL	PRIMERA
59	84	ALGEBRAICA	ALG	ZDR(2)	ZDR(2)	13:DRMWTRFL	PRIMERA
60	85	ALGEBRAICA	ALG	ZDR(3)	ZDR(3)	13:DRMWTRFL	PRIMERA
61	86	ALGEBRAICA	ALG	ZDR(4)	ZDR(4) ZDR(3)	13:DRMWTRFL	PRIMERA
62	87	ALGEBRAICA	ALG	ZDR(5)	ZDR(5) HDRS ZDR(1) HDRW ZDR(3) RDRW ZDR(2)	13:DRMWTRFL	PRIMERA
63	88	ALGEBRAICA	ALG	ZDR(6)	ZDR(6) QWWMW WDRS HDRS	13:DRMWTRFL	PRIMERA
64	89	ALGEBRAICA	ALG	ZDR(7)	ZDR(7) WDRS ZDR(5) ZDR(4) ZDR(6)	13:DRMWTRFL	PRIMERA
65	90	ALGEBRAICA	ALG	ZDR(8)	ZDR(8) ZDR(5) ZDR(4)	13:DRMWTRFL	PRIMERA
66	91	ALGEBRAICA	ALG	ZDR(9)	ZDR(9) ZDR(7) ZDR(8)	13:DRMWTRFL	PRIMERA
67	92	ALGEBRAICA	ALG	WFW	WFW ZDR(9)	13:DRMWTRFL	PRIMERA
68	93	ALGEBRAICA	ALG	PRHO	PRHO	14:RHTR	PRIMERA
69	94	ALGEBRAICA	ALG	TRHO	TRHO	14:RHTR	PRIMERA
70	95	ALGEBRAICA	ALG	PHPO	PHPO PRHO	14:RHTR	PRIMERA

No. Ecu. orden	No. Ecu. Desor.	Tipo de ecuación	Tipo Var.	Variable modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
71	96	ALGEBRAICA	ALG	WHPO	WHPO WHP	15:HPTRB	PRIMERA
72	97	ALGEBRAICA	ALG	NISEN	NISEN WHP	15:HPTRB	PRIMERA
73	98	ALGEBRAICA	ALG	HHPOI	HHPOI SSSO PHPO	15:HPTRB	PRIMERA
74	99	ALGEBRAICA	ALG	HHPO	HHPO HHPOI NISEN	15:HPTRB	PRIMERA
75	100	ALGEBRAICA	ALG	MWHP	MWHP WHP HHPO	15:HPTRB	PRIMERA
76	101	ALGEBRAICA	ALG	THPO	THPO PHPO HHPO	15:HPTRB	PRIMERA
77	102	ALGEBRAICA	ALG	HLPO	HLPO	16:CNDNSR	PRIMERA
78	103	ALGEBRAICA	ALG	WRHO	WRHO PRHO	17:LPTRB	PRIMERA
79	104	ALGEBRAICA	ALG	WLP	WLP WRHO	17:LPTRB	PRIMERA
80	105	ALGEBRAICA	ALG	MWLP	MWLP WLP HLPO	17:LPTRB	PRIMERA
81	106	ALGEBRAICA	ALG	MWGNPU	MWGNPU	18:TRBGNRTR	PRIMERA
82	107	ALGEBRAICA	ALG	MWGN	MWGN MWGNPU	18:TRBGNRTR	PRIMERA
83	108	ALGEBRAICA	ALG	MWTRO	MWTRO MWHP MWLP	18:TRBGNRTR	PRIMERA
84	109	ALGEBRAICA	ALG	HCPO	HCPO	19:CNDPMP	PRIMERA
85	110	ALGEBRAICA	ALG	HLHD	HLHD WSSO	20:LFPDHTR	PRIMERA
86	111	ALGEBRAICA	ALG	HDED	HDED WSSO	21:DRDR	PRIMERA
87	112	ALGEBRAICA	ALG	HHHD	HHHD WSSO	22:HPHTR	PRIMERA
88	113	ALGEBRAICA	ALG	HFPO	HFPO	23:BLRFDMPMP	PRIMERA
89	3	LIMITADOR	ALG	KTSSO	KTSSO WSSO	1:CONTSSO	SEGUNDA
90	4	TRANSDUCTOR	ALG	KCTSSO	KTSSO KCTSSO	1:CONTSSO	SEGUNDA
91	5	TRANSDUCTOR	ALG	CTSSO	CTSSO TSSO	1:CONTSSO	SEGUNDA
92	6	ALGEBRAICA	ALG	C1SC	C1SC KCTSSO CTSSO	1:CONTSSO	SEGUNDA
93	7	ALGEBRAICA	ALG	C2SC	C2SC C1SC	1:CONTSSO	SEGUNDA
94	8	LIMITADOR	ALG	C3SC	C3SC C2SC X1SC	1:CONTSSO	SEGUNDA
95	114	ALGEBRAICA	ALG	Y2SC	Y2SC X2SC	1:CONTSSO	SEGUNDA
96	115	TRANSDUCTOR	ALG	WSY	WSY Y2SC	1:CONTSSO	SEGUNDA
97	116	DERIVADA	DER	DX1SC	DX1SC C2SC	1:CONTSSO	SEGUNDA
98	117	DERIVADA	DER	DX2SC	DX2SC C3SC X2SC	1:CONTSSO	SEGUNDA
99	12	TRANSDUCTOR	ALG	CMWTRO	CMWTRO MWTRO	2:CONTRB	SEGUNDA
100	13	ALGEBRAICA	ALG	C1TR	C1TR CMWTRO	2:CONTRB	SEGUNDA
101	14	ALGEBRAICA	ALG	C2TR	C2TR C1TR X1TR	2:CONTRB	SEGUNDA
102	15	LIMITADOR	ALG	C3TR	C3TR C2TR	2:CONTRB	SEGUNDA
103	119	DERIVADA	DER	DX1TR	DX1TR C1TR	2:CONTRB	SEGUNDA
104	120	DERIVADA	DER	DX2TR	DX2TR C3TR X2TR	2:CONTRB	SEGUNDA
105	121	DERIVADA	DER	DX3TR	DX3TR C6TR X3TR	2:CONTRB	SEGUNDA

No. Ecu. orden	No. Ecu. Desor.	Tipo de ecuación	Tipo Var.	Variable modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
106	27	TRANSDUCTOR	ALG	CPSSO	CPSSO PSSO	3:CONCOMB	SEGUNDA
107	28	ALGEBRAICA	ALG	C1DM	C1DM KCPSO CPSSO	3:CONCOMB	SEGUNDA
108	29	ALGEBRAICA	ALG	C2DM	C2DM C1DM	3:CONCOMB	SEGUNDA
109	30	LIMITADOR	ALG	C3DM	C3DM C2DM X1DM	3:CONCOMB	SEGUNDA
110	31	ALGEBRAICA	ALG	C4DM	C4DM CNTR	3:CONCOMB	SEGUNDA
111	32	LIMITADOR	ALG	C5DM	C5DM C3DM C4DM	3:CONCOMB	SEGUNDA
112	33	ALGEBRAICA	ALG	C1AR	C1AR C5DM	3:CONCOMB	SEGUNDA
113	34	ALGEBRAICA	ALG	C2AR	C2AR C1AR X2AR	3:CONCOMB	SEGUNDA
114	35	ALGEBRAICA	ALG	C3AR	C3AR C2AR	3:CONCOMB	SEGUNDA
115	36	LIMITADOR	ALG	C4AR	C4AR C3AR X1AR	3:CONCOMB	SEGUNDA
116	37	ALGEBRAICA	ALG	C1CB	C1CB C5DM X2AR	3:CONCOMB	SEGUNDA
117	38	ALGEBRAICA	ALG	C2CB	C2CB C1CB X2CB	3:CONCOMB	SEGUNDA
118	39	ALGEBRAICA	ALG	C3CB	C3CB C2CB	3:CONCOMB	SEGUNDA
119	40	LIMITADOR	ALG	C4CB	C4CB C3CB X1CB	3:CONCOMB	SEGUNDA
120	124	DERIVADA	DER	DX1DM	DX1DM C2DM	3:CONCOMB	SEGUNDA
121	125	DERIVADA	DER	DX1AR	DX1AR C3AR	3:CONCOMB	SEGUNDA
122	126	DERIVADA	DER	DX2AR	DX2AR C4AR X2AR	3:CONCOMB	SEGUNDA
123	127	DERIVADA	DER	DX1CB	DX1CB C3CB	3:CONCOMB	SEGUNDA
124	128	DERIVADA	DER	DX2CB	DX2CB C4CB X2CB	3:CONCOMB	SEGUNDA
125	41	ALGEBRAICA	ALG	KCGR	KCGR XGG	4:CONTRHO	SEGUNDA
126	42	ALGEBRAICA	ALG	C1GR	C1GR X2RC	4:CONTRHO	SEGUNDA
127	48	LIMITADOR	ALG	KTRH	KTRH WSSO	4:CONTRHO	SEGUNDA
128	49	TRANSDUCTOR	ALG	KCTRH	KTRH KCTRH	4:CONTRHO	SEGUNDA
129	50	TRANSDUCTOR	ALG	CTRHO	CTRHO TRHO	4:CONTRHO	SEGUNDA
130	51	ALGEBRAICA	ALG	C1RC	C1RC KCTRH CTRHO	4:CONTRHO	SEGUNDA
131	52	ALGEBRAICA	ALG	C2RC	C2RC C1RC	4:CONTRHO	SEGUNDA
132	53	LIMITADOR	ALG	C3RC	C3RC C2RC X1RC	4:CONTRHO	SEGUNDA
133	132	DERIVADA	DER	DX1GR	DX1GR C1GR KCGR	4:CONTRHO	SEGUNDA
134	133	DERIVADA	DER	DX2GR	DX2GR X1GR X2GR	4:CONTRHO	SEGUNDA
135	134	ALGEBRAICA	ALG	Y3RC	Y3RC X3RC	4:CONTRHO	SEGUNDA
136	135	TRANSDUCTOR	ALG	WRY	WRY Y3RC	4:CONTRHO	SEGUNDA
137	137	DERIVADA	DER	DX1RC	DX1RC C2RC	4:CONTRHO	SEGUNDA
138	138	DERIVADA	DER	DX2RC	DX2RC C3RC X2RC	4:CONTRHO	SEGUNDA
139	139	DERIVADA	DER	DX3RC	DX3RC C3RC X3RC	4:CONTRHO	SEGUNDA
140	140	DERIVADA	DER	DHECO	DHECO WFW QEC	8:ECNMDR	SEGUNDA

No. Ecu. orden	No. Ecu. Desor.	Tipo de ecuación	Tipo Var.	Variable modificada	Dependencias	Módulo	Sección del módulo donde se calcula la ecuación.
141	141	ALGEBRAICA	ALG	ZDO(1)	ZDO(1)	9:DOMO	SEGUNDA
142	142	ALGEBRAICA	ALG	ZDO(2)	ZDO(2)	9:DOMO	SEGUNDA
143	143	ALGEBRAICA	ALG	ZDO(3)	ZDO(3)	9:DOMO	SEGUNDA
144	144	ALGEBRAICA	ALG	ZDO(4)	ZDO(4) ZDO(3)	9:DOMO	SEGUNDA
145	145	ALGEBRAICA	ALG	ZDO(5)	ZDO(5) HDRS ZDO(1) HDRW ZDO(3) RDRW ZDO(2)	9:DOMO	SEGUNDA
146	146	ALGEBRAICA	ALG	ZDO(6)	ZDO(6) QWWMW WDRS HDRS	9:DOMO	SEGUNDA
147	147	ALGEBRAICA	ALG	ZDO(7)	ZDO(7) WDRS ZDO(5) ZDO(4) ZDO(6)	9:DOMO	SEGUNDA
148	148	ALGEBRAICA	ALG	ZDO(8)	ZDO(8) ZDO(5) ZDO(4)	9:DOMO	SEGUNDA
149	149	ALGEBRAICA	ALG	ZDO(9)	ZDO(9) ZDO(7) ZDO(8)	9:DOMO	SEGUNDA
150	150	ALGEBRAICA	ALG	ZDO(10)	ZDO(10) ZDO(9) WDRS ZDO(4)	9:DOMO	SEGUNDA
151	151	DERIVADA	DER	DRDRS	DRDRS ZDO(10)	9:DOMO	SEGUNDA
152	152	DERIVADA	DER	DTWWM	DTWWM QWWMW QWWMW	10:PRDH20	SEGUNDA
153	153	DERIVADA	DER	DRSSO	DRSSO WDRS WSY WSSO	11:SPRTHTR	SEGUNDA
154	154	DERIVADA	DER	DHSSO	DHSSO WDRS HDRS WSY HFPO WSSO QSH	11:SPRTHTR	SEGUNDA
155	155	DERIVADA	DER	DRRHO	DRRHO WHPO WRY WRHO	14:RHTR	SEGUNDA
156	156	DERIVADA	DER	DHRHO	DHRHO WHPO HHPO WRY HFPO WRHO QRH	14:RHTR	SEGUNDA
157	157	DERIVADA	DER	DNTR	DNTR MWTR0 MWGN	18:TRBGNRTR	SEGUNDA
158	158	DERIVADA	DER	DDELTA	DDELTA	18:TRBGNRTR	SEGUNDA
159	159	DERIVADA	DER	DHIHO	DHIHO HCPO HLHD	20:LFPDHTR	SEGUNDA
160	160	DERIVADA	DER	DHDEW	DHDEW HDED	21:DRDR	SEGUNDA
161	161	DERIVADA	DER	DHHHO	DHHHO HFPO HHH	22:HPHTR	SEGUNDA

Tabla c.1 Secuencia de ejecución de las ecuaciones del modelo ordenado explícito.

Las ecuaciones que limitan el valor de una variable de estado si ésta se encuentra fuera de rangos permisibles, quedaron después del ordenamiento como sigue:

No. llamado: Nom. Mod	No. ec orden	No. ec des.	Tipo de ecuación	Tipo Var. que modif	Variable modificada	Rango permisible
1:CONTSSO	1	1	LIMITADOR	EST	X1SC	$1 \leq \text{valor} \leq 5$
1:CONTSSO	2	2	LIMITADOR	EST	X2SC	$1 \leq \text{valor} \leq 5$
2:CONTRB	3	9	LIMITADOR	EST	X1TR	$-1 \leq \text{valor} \leq 1$
2:CONTRB	4	10	LIMITADOR	EST	X2TR	$0 \leq \text{valor} \leq 5$
2:CONTRB	5	11	LIMITADOR	EST	X3TR	$1 \leq \text{valor} \leq 5$
3:CONCOMB	11	20	LIMITADOR	EST	X1DM	$1 \leq \text{valor} \leq 5$
3:CONCOMB	12	21	LIMITADOR	EST	X1AR	$1 \leq \text{valor} \leq 5$
3:CONCOMB	13	22	LIMITADOR	EST	X2AR	$1 \leq \text{valor} \leq 5$
3:CONCOMB	14	23	LIMITADOR	EST	X1CB	$1 \leq \text{valor} \leq 5$
3:CONCOMB	15	24	LIMITADOR	EST	X2CB	$1 \leq \text{valor} \leq 5$
4:CONTRHO	20	43	LIMITADOR	EST	X1GR	$1 \leq \text{valor} \leq 5$
4:CONTRHO	23	44	LIMITADOR	EST	X2GR	$1 \leq \text{valor} \leq \text{KCWGR} \leq 5$
4:CONTRHO	25	45	LIMITADOR	EST	X1RC	$1 \leq \text{valor} \leq 5$
4:CONTRHO	26	46	LIMITADOR	EST	X2RC	$1 \leq \text{valor} \leq 5$
4:CONTRHO	27	47	LIMITADOR	EST	X3RC	$1 \leq \text{valor} \leq 5$

Las ecuaciones diferenciales después del ordenamiento quedaron en las siguientes posiciones:

No. llamado: Nom. Mod	No. ec. orden	No. ec des	Tipo de ecuación	Tipo Var que modif	Variable modificada	Variable de estado correspondiente
1:CONTSSO	97	116	DERIVADA	DER	DX1SC	X1SC
1:CONTSSO	98	117	DERIVADA	DER	DX2SC	X2SC
2:CONTRB	103	119	DERIVADA	DER	DX1TR	X1TR
2:CONTRB	104	120	DERIVADA	DER	DX2TR	X2TR
2:CONTRB	105	121	DERIVADA	DER	DX3TR	X3TR
3:CONCOMB	120	124	DERIVADA	DER	DX1DM	X1DM
3:CONCOMB	121	125	DERIVADA	DER	DX1AR	X1AR

No. llamado: Nom. Mod	No. ec. orden	No. ec des	Tipo de ecuación	Tipo Var que modif	Variable modificada	Variable de estado correspondiente
3:CONCOMB	122	126	DERIVADA	DER	DX2AR	X2AR
3:CONCOMB	123	127	DERIVADA	DER	DX1CB	X1CB
3:CONCOMB	124	128	DERIVADA	DER	DX2CB	X2CB
4:CONTRHO	133	132	DERIVADA	DER	DX1GR	X1GR
4:CONTRHO	134	133	DERIVADA	DER	DX2GR	X2GR
4:CONTRHO	137	137	DERIVADA	DER	DX1RC	X1RC
4:CONTRHO	138	138	DERIVADA	DER	DX2RC	X2RC
4:CONTRHO	139	139	DERIVADA	DER	DX3RC	X3RC
8:ECNMDR	140	140	DERIVADA	DER	DHECO	HECO
9:DOMO	151	151	DERIVADA	DER	DRDRS	RDRS
10:PRDH2O	152	152	DERIVADA	DER	DTWWM	TWWM
11:SPRTHTR	153	153	DERIVADA	DER	DRSSO	DRSSO
11:SPRTHTR	154	154	DERIVADA	DER	DHSSO	HSSO
14:RHTR	155	155	DERIVADA	DER	DRRHO	RRHO
14:RHTR	156	156	DERIVADA	DER	DHRHO	HRHO
18:TRBGNRTR	157	157	DERIVADA	DER	DNTR	NTR
18:TRBGNRTR	158	158	DERIVADA	DER	DDELTA	DELTA
20:LPHDHTR	159	159	DERIVADA	DER	DH1HO	H1HO
21:DRDR	160	160	DERIVADA	DER	DHDEW	HDEW
22:HPHTR	161	161	DERIVADA	DER	DHHHO	HHHO

64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 79
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88
89 89
90 90
91 91
92 92
93 93
94 94
95 95
96 96
97 97
98 98
99 99
100 100
101 101
102 102
103 103
104 104
105 105
106 106
107 107
108 108
109 109
110 110
111 111
112 112
113 113
114 114
115 115
116 116
117 117
118 118
119 119
120 120
121 121
122 122
123 123
124 124
125 125
126 126
127 127
128 128
129 129
130 130
131 131
132 132
133 133
134 134
135 135
136 136
137 137
138 138
139 139
140 140
141 141
142 142
143 143
144 144
145 145
146 146
147 147
148 148
149 149
150 150
151 151
152 152
153 153
154 154
155 155
156 156
157 157
158 158
159 159
160 160
161 161

APÉNDICE E. MATRIZ ESTRUCTURAL FINAL, ORDENAMIENTO IMPLANTADO

EQU.	VAR.
1	1
2	1
3	1
4	1
5	1
6	1
7	11
8	11
9	1 11
10	1 1
11	1
12	1
13	1
14	1
15	1
16	1
17	11
18	1 1
19	1 1
20	1
21	1 1
22	11
23	11
24	11
25	1
26	1
27	1 1
28	1 1
29	11 1 1
30	11 1 1
31	111
32	1 1
33	1 1
34	111
35	1 11
36	1 1 111
37	1 1 11
38	1 111
39	1 11
40	1 1 11
41	1 1 1
42	1 1 1
43	1 1 1 1
44	1 1 1 1
45	1 1 11
46	1
47	1
48	1
49	1
50	1
51	1 1
52	1
53	1 11
54	1
55	1
56	1 1 1
57	1 1
58	1
59	1
60	1
61	1 11
62	1 1 111 1
63	1 1 1 1

64 64 1. 1111.
65 65 11. 1.
66 66 111.
67 67 11.
68 68 1.
69 69 1.
70 70 1. 1.
71 71 1. 1.
72 72 1. 1.
73 73 1. 1. 1.
74 74 111.
75 75 1. 1.
76 76 1. 1.
77 77 1.
78 78 1.
79 79 1.
80 80 1. 1.
81 81 11.
82 82 11.
83 83 1. 1.
84 84 1.
85 85 1.
86 86 1.
87 87 1.
88 88 1.
89 89 1.
90 90 11.
91 91 1. 111.
92 92 111.
93 93 11.
94 94 1.
95 95 1.
96 96 11.
97 97 1. 1.
98 98 1.
99 99 1. 1. 1.
100 100 11.
101 101 1.
102 102 11.
103 103 11.
104 104 1. 1.
105 105 1. 1.
106 106 1.
107 107 1. 11.
108 108 11.
109 109 1.
110 110 1.
111 111 111.
112 112 1.
113 113 1.
114 114 11.
115 115 1. 11.
116 116 1. 1. 1.
117 117 1. 11.
118 118 11.
119 119 1. 11.
120 120 1. 1.
121 121 1. 1. 1.
122 122 1. 1. 1.
123 123 1. 1. 1.
124 124 1. 1. 1.
125 125 1.
126 126 1.
127 127 1.
128 128 11.
129 129 1.
130 130 111.
131 131 11.
132 132 1.
133 133 11. 1.
134 134 1. 1.
135 135 1.
136 136 1. 11.
137 137 1. 1.
138 138 1. 1. 1.
139 139 1. 1. 1.
140 140 1.
141 141 1.
142 142 1.
143 143 1.
144 144 11.
145 145 11. 1.
146 146 11. 1.
147 147 1111.
148 148 11. 1.
149 149 111.
150 150 1. 11.
151 151 11.
152 152 1.
153 153 1.
154 154 11. 1. 1.
155 155 1. 1.
156 156 1. 1. 1.
157 157 11.
158 158 1.
159 159 11.
160 160 1.
161 161 11.

APÉNDICE D. MATRIZ ESTRUCTURAL FINAL, LEDET

EQN.	VAR.	
1	1	1
2	2	1
9	9	1
10	10	1
11	11	1
16	16	1
17	17	1
18	18	1
19	19	1
20	20	1
21	21	1
22	22	1
23	23	1
24	24	1
25	25	1
26	26	1
31	31	1
43	43	1
45	45	1
46	46	1
42	42	1
47	47	1
71	71	1
72	72	1
73	73	1
74	74	1
75	75	1
76	76	1
77	77	1
27	27	1
28	28	1
29	29	1
30	30	1
32	32	1
33	33	1
34	34	1
35	35	1
36	36	1
37	37	1
38	38	1
39	39	1
40	40	1
78	78	1
79	79	1
5	5	1
80	80	1
83	83	1
84	84	1
85	85	1
86	86	1
87	87	1
88	88	1
89	89	1
90	90	1
91	91	1
92	92	1
93	93	1
94	94	1
50	50	1
95	95	1
98	96	1
102	102	1
103	103	1
104	104	1
105	105	1
106	106	1
107	107	1
109	109	1

113 113
114 114
115 115
118 118
81 81
3 3
4 4
6 6
7 7
8 8
48 48
49 49
51 51
52 52
53 53
82 82
96 96
97 97
99 99
100 100
101 101
108 108
12 12
13 13
14 14
15 15
110 110
111 111
112 112
116 116
117 117
119 119
120 120
121 121
122 122
123 123
58 58
124 124
125 125
126 126
127 127
128 128
129 129
130 130
44 44
131 131
54 54
55 55
56 56
65 65
66 66
67 67
133 133
134 134
135 135
136 136
41 41
57 57
59 59
60 60
61 61
62 62
63 63
64 64
68 68
69 69
70 70
132 132
137 137
138 138
139 139
140 140
141 141
142 142
143 143
144 144
145 145
146 146
147 147
148 148
149 149
150 150
151 151
152 152
153 153
154 154
155 155
156 156
157 157
158 158
159 159
160 160
161 161

APÉNDICE F

Se presenta la validación del nuevo ordenamiento del modelo.

F.1 Prueba de Consistencia Aplicada al Modelo Ordenado.

Para tener un margen de confianza de que el ordenamiento realizado a las ecuaciones del modelo se realizó en forma correcta, se aplicó al modelo ordenado la Prueba de Consistencia [Ojeda_2].

Como se mencionó en el capítulo III en la sección III.1.3, ésta prueba consiste en hacer dos evaluaciones del modelo en un mismo tiempo dado, la primera vez se evalúa con el conjunto inicial de datos que se tienen del modelo en dicho tiempo y la segunda vez se evalúa el modelo con los valores obtenidos de la primera evaluación, realizando posteriormente una comparación entre los resultados obtenidos en las dos evaluaciones. Si la diferencia no es nula entonces se tienen problemas de ordenamiento y/o variables que requieren de un valor de inicio.

La prueba de consistencia se aplicó al modelo ordenado con la simulación del modelo en estado transitorio en una de las pruebas seleccionada, como se explicó en el capítulo III, sección III.2.2.

Se eligieron 16 tiempos para la aplicación de la prueba de consistencia al modelo ordenado los cuales se presentan en la siguiente tabla.

No	Tiempo elegido	No	Tiempo elegido	No	Tiempo elegido	No	Tiempo elegido
1	0 s	5	45 s	9	95 s	13	140 s
2	5 s	6	60 s	10	110 s	14	150 s
3	15 s	7	75 s	11	120 s	15	200 s
4	30 s	8	90 s	12	130 s	16	380 s

F.1.1 Resultados de la prueba.

La diferencia reportada en los 16 tiempos elegidos fue nula, este resultado indica que la prueba de consistencia fue aprobada y que las ecuaciones del modelo tienen una secuencia de ejecución correcta.

F.2 Prueba de Estabilidad del Modelo Ordenado

F.2.1 Valores Característicos del Modelo

Los valores característicos del modelo en un tiempo dado, son los valores característicos de la matriz jacobiana obtenida en dicho tiempo del sistema de ecuaciones diferenciales. Los valores característicos del modelo influyen en el buen o mal funcionamiento de la fórmula BDF de orden 1 para la obtención de la solución de las ecuaciones diferenciales. Dependiendo de si éstos se encuentran o no dentro de su zona de inestabilidad (el interior y frontera del círculo unitario con centro en $(1,0)$ del plano complejo), ver gráfica 2.2.1 en el capítulo II sección II.2

Debido a lo anterior, se volvió a aplicar la misma simulación del modelo ordenado en estado transitorio elegida para la prueba de consistencia y se volvieron a elegir los mismos tiempos de la aplicación de la prueba de consistencia, para calcular las matrices jacobianas del sistema de ecuaciones diferenciales y poder obtener con ellas los valores característicos del modelo.

Para cada matriz jacobiana se obtuvieron sus valores característicos por medio del paquete SUPEIS [Ojeda_2].

En todas las matrices jacobianas se obtuvieron al menos 3 valores característicos igual a $0 + 0i$, éste valor característico se encuentra dentro de la zona de inestabilidad absoluta de la fórmula BDF de orden 1.

En las matrices jacobianas calculadas en los tiempos $t = 90, 95, 110, 120, 130$ y 140 segundos se obtuvo en cada caso, un valor característico con parte real positiva de orden de 10^{-2} o 10^{-3} . Estos valores característicos se encuentran dentro de la zona de inestabilidad absoluta de la fórmula BDF de orden 1 y se presentan cuando el modelo está en el estado

transitorio. Una vez que el modelo pasa el transitorio y vuelve a alcanzar el estado estacionario, estos valores característicos desaparecen.

A continuación se presentan los valores característicos obtenidos en cada una de las 16 matrices, los valores característicos se presentan en forma ascendente con respecto a la parte real de los valores característicos y se presentan a través de una tabla.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98514	0.91464	10	-6.01E-02	0	19	-4.17E-03	0
2	-0.98514	-0.91464	11	-1.72E-02	7.35E-04	20	-2.74E-03	0
3	-0.78745	0	12	-1.72E-02	-7.35E-04	21	-2.56E-03	6.46E-03
4	-0.75519	0.223854	13	-1.63E-02	0	22	-2.56E-03	-6.46E-03
5	-0.75519	-0.22385	14	-1.27E-02	0	23	0	0
6	-0.26417	0	15	-1.02E-02	1.802932	24	0	0
7	-0.20002	0	16	-1.02E-02	-1.80293	25	0	0
8	-7.91E-02	0.139407	17	-7.72E-03	0	26	0	0
9	-7.91E-02	-0.13941	18	-5.13E-03	0	27	0	0

Tabla f.2.1 Valores Característicos del modelo al inicio de la simulación.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.99991	0	10	-4.45E-02	-0.19579	19	-5.10E-03	0
2	-0.98338	0.916128	11	-1.71E-02	6.86E-04	20	-4.04E-03	2.34E-03
3	-0.98338	-0.91613	12	-1.71E-02	-6.86E-04	21	-4.04E-03	-2.34E-03
4	-0.68364	0.332319	13	-1.64E-02	0	22	-1.74E-03	0
5	-0.68364	-0.33232	14	-1.06E-02	1.57E-03	23	-2.93E-04	0
6	-0.26385	0	15	-1.06E-02	-1.57E-03	24	-2.93E-04	0
7	-0.20002	0	16	-1.03E-02	1.80354	25	0	0
8	-5.95E-02	0	17	-1.03E-02	-1.80354	26	0	0
9	-4.45E-02	0.195789	18	-5.80E-03	0	27	0	0

Tabla f.2.2 Valores Característicos del modelo obtenidos en el tiempo t = 5 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98454	0.914546	10	-6.11E-02	0	19	-6.20E-03	1.00E-03
2	-0.98454	-0.91455	11	-2.65E-02	0	20	-6.20E-03	-1.00E-03
3	-0.78718	0	12	-1.73E-02	0	21	-5.10E-03	0
4	-0.75547	0.223892	13	-1.61E-02	0	22	-2.47E-03	0
5	-0.75547	-0.22389	14	-1.43E-02	0	23	-2.93E-04	0
6	-0.26454	0	15	-1.27E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-1.02E-02	1.80665	25	0	0
8	-6.78E-02	0.129315	17	-1.02E-02	-1.80665	26	0	0
9	-6.78E-02	-0.12931	18	-7.31E-03	0	27	0	0

Tabla f.2.3 Valores Característicos del modelo obtenidos en el tiempo t = 15 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98197	0.92044	10	-6.14E-02	0	19	-5.42E-03	2.60E-03
2	-0.98197	-0.92044	11	-2.56E-02	0	20	-5.42E-03	-2.60E-03
3	-0.78741	0	12	-1.72E-02	0	21	-5.10E-03	0
4	-0.76426	0.237083	13	-1.62E-02	0	22	-2.72E-03	0
5	-0.76426	-0.23708	14	-1.45E-02	0	23	-2.93E-04	0
6	-0.26446	0	15	-1.27E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-1.03E-02	1.818101	25	0	0
8	-7.32E-02	0.129957	17	-1.03E-02	-1.8181	26	0	0
9	-7.32E-02	-0.12996	18	-7.13E-03	0	27	0	0

Tabla f.2.4 Valores Característicos del modelo obtenidos en el tiempo $t = 30$ segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98043	0.922517	10	-6.13E-02	0	19	-5.10E-03	0
2	-0.98043	-0.92252	11	-2.53E-02	0	20	-5.09E-03	2.79E-03
3	-0.78744	0	12	-1.72E-02	0	21	-5.09E-03	-2.79E-03
4	-0.77388	0.243203	13	-1.62E-02	0	22	-2.78E-03	0
5	-0.77388	-0.2432	14	-1.45E-02	0	23	-2.93E-04	0
6	-0.26285	0	15	-1.27E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-1.03E-02	1.831175	25	0	0
8	-7.56E-02	0.12996	17	-1.03E-02	-1.83118	26	0	0
9	-7.56E-02	-0.12996	18	-6.78E-03	0	27	0	0

Tabla f.2.5 Valores Característicos del modelo obtenidos en el tiempo $t = 45$ segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.97875	0.924346	10	-6.13E-02	0	19	-5.10E-03	0
2	-0.97875	-0.92435	11	-2.51E-02	0	20	-4.77E-03	2.86E-03
3	-0.78716	0	12	-1.71E-02	0	21	-4.77E-03	-2.86E-03
4	-0.78383	0.24846	13	-1.62E-02	0	22	-2.83E-03	0
5	-0.78383	-0.24846	14	-1.45E-02	0	23	-2.93E-04	0
6	-0.26083	0	15	-1.27E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-1.02E-02	1.842209	25	0	0
8	-7.73E-02	0.130282	17	-1.02E-02	-1.84221	26	0	0
9	-7.73E-02	-0.13028	18	-6.44E-03	0	27	0	0

Tabla f.2.6 Valores Característicos del modelo obtenidos en el tiempo $t = 60$ segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.97829	0.907208	10	-3.78E-02	-0.17982	19	-5.00E-03	0
2	-0.97829	-0.90721	11	-2.93E-02	0	20	-4.23E-03	0
3	-0.82726	0.241915	12	-1.70E-02	0	21	-1.68E-03	0
4	-0.82726	-0.24192	13	-1.64E-02	0	22	-2.93E-04	0
5	-0.78731	0	14	-1.27E-02	0	23	-2.93E-04	0
6	-0.25585	0	15	-1.21E-02	0	24	0	0
7	-0.20002	0	16	-9.66E-03	1.857489	25	0	0
8	-6.29E-02	0	17	-9.66E-03	-1.85749	26	0	0
9	-3.78E-02	0.179816	18	-6.18E-03	0	27	0	0

Tabla f.2.7 Valores Característicos del modelo obtenidos en el tiempo t = 75 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98026	0.867253	10	-5.67E-02	0	19	-5.17E-03	-6.19E-04
2	-0.98026	-0.86725	11	-2.85E-02	0	20	-3.37E-03	0
3	-0.8106	0	12	-1.70E-02	0	21	-2.93E-04	0
4	-0.79204	0.237657	13	-1.64E-02	0	22	-2.93E-04	0
5	-0.79204	-0.23766	14	-1.53E-02	1.63E-03	23	0	0
6	-0.24661	0	15	-1.53E-02	-1.63E-03	24	0	0
7	-0.20002	0	16	-8.43E-03	1.87177	25	0	0
8	-6.14E-02	0.158306	17	-8.43E-03	-1.87177	26	0	0
9	-6.14E-02	-0.15831	18	-5.17E-03	6.19E-04	27	3.42E-02	0

Tabla f.2.8 Valores Característicos del modelo obtenidos en el tiempo t = 90 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98073	0.855215	10	-5.60E-02	0	19	-5.09E-03	-6.41E-04
2	-0.98073	-0.85521	11	-3.01E-02	0	20	-3.37E-03	0
3	-0.8041	0	12	-1.70E-02	0	21	-2.93E-04	0
4	-0.7828	0.234666	13	-1.64E-02	0	22	-2.93E-04	0
5	-0.7828	-0.23467	14	-1.53E-02	1.68E-03	23	0	0
6	-0.24389	0	15	-1.53E-02	-1.68E-03	24	0	0
7	-0.20002	0	16	-8.09E-03	1.875523	25	0	0
8	-6.07E-02	0.152341	17	-8.09E-03	-1.87552	26	0	0
9	-6.07E-02	-0.15234	18	-5.09E-03	6.41E-04	27	2.77E-02	0

Tabla f.2.9 Valores Característicos del modelo obtenidos en el tiempo t = 95 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-1.05453	0	10	-5.18E-02	0	19	-4.63E-03	6.43E-04
2	-0.97996	0.822078	11	-5.00E-02	0	20	-4.63E-03	-6.43E-04
3	-0.97996	-0.82208	12	-1.98E-02	0	21	-3.37E-03	0
4	-0.6262	0.341406	13	-1.71E-02	0	22	-2.93E-04	0
5	-0.6262	-0.34141	14	-1.63E-02	0	23	-2.93E-04	0
6	-0.23811	0	15	-1.54E-02	1.93E-03	24	0	0
7	-0.20002	0	16	-1.54E-02	-1.93E-03	25	0	0
8	-7.14E-02	0.135542	17	-7.31E-03	1.877709	26	0	0
9	-7.14E-02	-0.13554	18	-7.31E-03	-1.87771	27	2.61E-02	0

Tabla f.2.10 Valores Característicos del modelo obtenidos en el tiempo t = 110 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-1.05534	0	10	-5.01E-02	0	19	-4.92E-03	0
2	-0.98073	0.814322	11	-5.00E-02	0	20	-3.26E-03	0
3	-0.98073	-0.81432	12	-2.32E-02	0	21	-2.34E-03	0
4	-0.61534	0.324349	13	-1.71E-02	0	22	-2.93E-04	0
5	-0.61534	-0.32435	14	-1.63E-02	0	23	-2.93E-04	0
6	-0.23859	0	15	-1.52E-02	2.05E-03	24	0	0
7	-0.20002	0	16	-1.52E-02	-2.05E-03	25	0	0
8	-5.83E-02	0.114296	17	-7.16E-03	1.875429	26	0	0
9	-5.83E-02	-0.1143	18	-7.16E-03	-1.87543	27	7.49E-03	0

Tabla f.2.11 Valores Característicos del modelo obtenidos en el tiempo t = 120 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-1.05072	0	10	-5.08E-02	0	19	-5.07E-03	0
2	-0.98031	0.818134	11	-5.00E-02	0	20	-3.47E-03	0
3	-0.98031	-0.81813	12	-2.01E-02	0	21	-1.65E-03	0
4	-0.61284	0.325744	13	-1.71E-02	0	22	-2.93E-04	0
5	-0.61284	-0.32574	14	-1.63E-02	0	23	-2.93E-04	0
6	-0.24044	0	15	-1.52E-02	2.30E-03	24	0	0
7	-0.20002	0	16	-1.52E-02	-2.30E-03	25	0	0
8	-6.37E-02	0.127858	17	-7.27E-03	1.872793	26	0	0
9	-6.37E-02	-0.12786	18	-7.27E-03	-1.87279	27	1.37E-02	0

Tabla f.2.12 Valores Característicos del modelo obtenidos en el tiempo t = 130 segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-1.04547	0	10	-5.22E-02	0	19	-4.74E-03	0
2	-0.97979	0.827665	11	-5.00E-02	0	20	-4.12E-03	0
3	-0.97979	-0.82766	12	-1.83E-02	0	21	-3.17E-03	0
4	-0.6146	0.338077	13	-1.72E-02	0	22	-2.93E-04	0
5	-0.6146	-0.33808	14	-1.63E-02	0	23	-2.93E-04	0
6	-0.24268	0	15	-1.53E-02	2.30E-03	24	0	0
7	-0.20002	0	16	-1.53E-02	-2.30E-03	25	0	0
8	-7.57E-02	0.152322	17	-7.48E-03	1.871732	26	0	0
9	-7.57E-02	-0.15232	18	-7.48E-03	-1.87173	27	3.15E-02	0

Tabla f.2.13 Valores Característicos del modelo obtenidos en el tiempo $t = 140$ segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98259	0.837288	10	-5.43E-02	0	19	-5.77E-03	0
2	-0.98259	-0.83729	11	-5.00E-02	0	20	-4.78E-03	2.70E-03
3	-0.78737	0	12	-2.50E-02	0	21	-4.78E-03	-2.70E-03
4	-0.73511	0.226117	13	-1.71E-02	0	22	-2.84E-03	0
5	-0.73511	-0.22612	14	-1.62E-02	0	23	-2.93E-04	0
6	-0.243	0	15	-1.45E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-1.27E-02	0	25	0	0
8	-6.56E-02	0.132573	17	-7.64E-03	1.87032	26	0	0
9	-6.56E-02	-0.13257	18	-7.64E-03	-1.87032	27	0	0

Tabla f.2.14 Valores Característicos del modelo obtenidos en el tiempo $t = 150$ segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.982	0.84085	10	-5.46E-02	0	19	-5.74E-03	0
2	-0.982	-0.84085	11	-5.00E-02	0	20	-4.63E-03	2.82E-03
3	-0.78731	0	12	-2.48E-02	0	21	-4.63E-03	-2.82E-03
4	-0.73937	0.229855	13	-1.71E-02	0	22	-2.89E-03	0
5	-0.73937	-0.22985	14	-1.63E-02	0	23	-2.93E-04	0
6	-0.2445	0	15	-1.45E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-1.27E-02	0	25	0	0
8	-6.63E-02	0.133441	17	-7.72E-03	1.871392	26	0	0
9	-6.63E-02	-0.13344	18	-7.72E-03	-1.87139	27	0	0

Tabla f.2.15 Valores Característicos del modelo obtenidos en el tiempo $t = 200$ segundos.

No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria	No.	Parte real	Parte imaginaria
1	-0.98099	0.844953	10	-5.53E-02	0	19	-5.72E-03	0
2	-0.98099	-0.84495	11	-2.46E-02	0	20	-4.46E-03	2.89E-03
3	-0.78726	0	12	-1.71E-02	0	21	-4.46E-03	-2.89E-03
4	-0.74243	0.231758	13	-1.63E-02	0	22	-2.93E-03	0
5	-0.74243	-0.23176	14	-1.46E-02	0	23	-2.93E-04	0
6	-0.24735	0	15	-1.27E-02	0	24	-2.93E-04	0
7	-0.20002	0	16	-9.79E-03	0	25	0	0
8	-6.70E-02	0.133479	17	-7.83E-03	1.871775	26	0	0
9	-6.70E-02	-0.13348	18	-7.83E-03	-1.87178	27	0	0

Tabla f.2.16 Valores Característicos del modelo obtenidos en el tiempo $t = 380$ segundos.

F.3 Solución y Validación del Modelo Ordenado

Se aplicaron las pruebas descritas en la sección III.2 al modelo original y al modelo ordenado, utilizando como método integrador el Runge-Kutta de cuarto orden con un paso de integración de 0.1 seg. y se obtuvieron muestras de los resultados obtenidos en cada segundo de simulación de las 27 variables de estado y de las 42 variables algebraicas que son salidas a los tableros del simulador.

Se compararon los resultados obtenidos por el modelo ordenado con los resultados obtenidos por el modelo original, midiendo la diferencia cometida por el modelo ordenado respecto del modelo original, ésta diferencia se midió como error relativo si el valor de la variable reportado por el modelo original era mayor que 1 y como error absoluto cuando el valor era menor o igual a 1. En las tablas f.3.1 y f.3.2 se presentan los máximos errores cometidos en las 27 variables de estado y en las 42 variables algebraicas en la aplicación de las pruebas. En la tabla f.3.1 se aprecia que en las variables de estado el error más grande que se cometió en la prueba del modelo en estado estacionario se cometió en la variable de X2AR con un error de 0.1769 % y que en la prueba del modelo en estado transitorio el error más grande se cometió en la variable X2CB con un error de 0.3101%. En la tabla f.3.2 se aprecia que en las variables algebraicas de salida a los tableros del simulador el error más grande obtenido en la prueba del estado estacionario se obtuvo en la variable QEC con un error del 0.2633% y para la prueba del modelo en estado transitorio se obtuvo en la variable QSH con un error de 0.7728%.

VARIABLE DE ESTADO	PRUEBA 1. ESTACIONARIO		PRUEBA 2. TRANSITORIO	
	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)
X1SC	0.	0.	0.	0.
X2SC	0.	0.	0.	0.
X1TR	0.	0.	0.2463	158.0
X2TR	0.1579E-01	118.0	0.4113E-01	155.0
X3TR	0.1580E-01	66.00	0.4115E-01	156.0
X1DM	0.9097E-01	48.00	0.1026	133.0
X1AR	0.7424E-01	43.00	0.9726E-01	131.0
X2AR	0.1769	28.00	0.2161	95.00
X1CB	0.6227E-01	59.00	0.7087E-01	132.0
X2CB	0.1633	1.000	0.3101	76.00
X1GR	0.	0.	0.	0.
X2GR	0.	0.	0.2842	95.00
X1RC	0.	0.	0.	0.
X2RC	0.	0.	0.	0.
X3RC	0.	0.	0.	0.
HECO	0.1482E-01	111.0	0.1192E-01	140.0
RDRS	0.2308E-01	18.00	0.3881E-01	90.00
TWWM	0.	0.	0.8872E-02	98.00
RSSO	0.2197E-01	101.0	0.2675E-01	131.0
HSSO	0.1291E-01	41.00	0.1345E-01	131.0
RRHO	0.3118E-01	56.00	0.3453E-01	164.0
HRHO	0.7029E-02	26.00	0.1449E-01	143.0
NTR	0.	0.	0.	0.
DELTA	0.7318E-02	14.00	0.1504	43.00
HIHO	0.	0.	0.5274E-02	396.0
HDEW	0.3989E-02	124.0	0.4068E-02	374.0
HHHO	0.2246E-02	23.00	0.2321E-02	377.0

Tabla f.3.1 Máximos errores cometidos en las variables de estado en la validación del ordenamiento del modelo. (Modelo ordenado vs Modelo Original, método integrador RK4, h=0.1 seg.).

VARIABLE ALGEBRAICA SALIDA A LOS TABLEROS	PRUEBA 1. ESTACIONARIO		PRUEBA 2. TRANSITORIO	
	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)	MAX. ERROR COMETIDO (%)	TIEMPO EN QUE OCURRIÓ (s)
WG	0.1616	25.00	0.4117	96.00
TFN1	0.8595E-01	1.000	0.3487	77.00
TWWGO	0.7218E-01	1.000	0.2659	77.00
QWWGM	0.2434	1.000	0.7556	77.00
HDRS	0.9266E-02	7.000	0.9267E-02	27.00
TDRS	0.	0.	0.8942E-02	117.0
PDRS	0.1156E-01	9.000	0.2548E-01	92.00
HDRW	0.4058E-02	10.00	0.8475E-02	95.00
RDRW	0.8831E-02	9.000	0.1394E-01	88.00
WDRS	0.2156E-01	52.00	0.6214E-01	84.00
PSSO	0.1242E-01	14.00	0.2336E-01	147.0
SSSO	0.1253E-01	58.00	0.6432E-02	109.0
TSSO	0.1883E-01	48.00	0.2008E-01	138.0
QWWMW	0.7651E-01	3.000	0.2267	80.00
WFW	0.3098E-01	2.000	0.4511E-01	95.00
WSSO	0.1425E-01	52.00	0.2242E-01	165.0
WSY	0.	0.	0.	0.
ACV	0.1580E-01	114.0	0.4110E-01	154.0
WRY	0.	0.	0.	0.
WAR	0.2366	26.00	0.3862	96.00
WGR	0.	0.	0.3862	96.00
WFL	0.2149	1.000	0.6028	77.00
XGG	0.	0.	0.	0.
QSH	0.2101	1.000	0.7728	95.00
QRH	0.1826	1.000	0.5429	77.00
QEC	0.2633	30.00	0.5188	77.00
PHPO	0.1070E-01	55.00	0.2434E-01	166.0
PRHO	0.1420E-01	101.0	0.2382E-01	170.0
TRHO	0.1572E-01	42.00	0.2531E-01	141.0
WHPO	0.1459E-01	70.00	0.2337E-01	163.0
HHPO	0.8736E-02	32.00	0.9186E-02	107.0
THPO	0.1494E-01	48.00	0.2095E-01	130.0
MWHP	0.1133E-01	160.0	0.1976E-01	94.00
WRHO	0.1327E-01	57.00	0.2335E-01	167.0
MWLP	0.2044E-01	46.00	0.3590E-01	134.0
HFPO	0.3822E-02	124.0	0.3894E-02	375.0
HCPO	0.	0.	0.	0.
MWGN	0.6521E-02	11.00	0.1370	43.00
HLHD	0.	0.	0.7607E-02	92.00
HDED	0.3563E-02	44.00	0.5608E-02	164.0
HHHD	0.5434E-02	25.00	0.5787E-02	100.00
MWTRO	0.1302E-01	90.00	0.2038E-01	138.0

Tabla f.3.2 Máximos errores cometidos en las variables algebraicas de salida a tableros, en la validación del ordenamiento del modelo. (Modelo ordenado vs Modelo Original, método integrador RK4, h=0.1 seg.).

Bibliografía

- [ANSI] ANSI/ANS-3.5-1981, *American National Standard Nuclear Power Plant Simulators for use in Operator Training*.
- [Boston] Boston J. F. et al, *Computing in 2001 Software:Tackling Touger Task*, Chemical Engineering Progress, Noviembre 1993, pp. 38-49.
- [Brayton] Brayton R.K. et al, *A New Efficient Algorithm for Solving Differential-Algebraic Systems Using Implicit Backward Differentiation Formulas*, Proceedings of the IEEE, Vol. 60, No. 3, Enero 1972, pp. 95-108.
- [Brenan] Brenan K. E. et al, *Numerical Solution of Initial-Value Problems in Differential-Algebraic equations*, North-Holland, 1989.
- [Cameron] Cameron I. T., *Numerical Solution of Differential-Algebraic Equations in Process Dynamics*, Tesis Doctoral, Department of Chemical Engineering and Chemical Technology, Imperial College of Science and Technology, London SW7, 1981.
- [Castelazo] Castelazo I. A., *Simulador de Plantas Generadoras para Entrenamiento de Operadores*, Rpte. Interno del IIE, Cuernavaca, Mor.,1983.
- [Conte] Conte S. D., *Elementary Numerical Analysis*, McGraw-Hill, 1980.
- [Courant] Courant R., *Introducción al Cálculo y al Análisis Matemático Vol. 1*, Limusa, 1989.
- [Creus] Creus A. S., *Simulación de Procesos con PC*, Marcombo, 1989.
- [Dongarra] Dongarra J. R et al, *LINPACK User's Guide*, SIAM, 1979.
- [Forsythe] Forsythe G. E. et al, *Computer Methods for Mathematical Computation*, Prentice Hall, 1977, pp. 48-55.
- [Gear_1] Gear C. W., *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1971, pp.215-216
- [Gear_2] Gear C. W., *Simultaneous Numerical Solution of Differential-Algebraic Equations*, IEEE Transactions on Circuit Theory, Vol. CT-18, No. 1, Enero 1971, pp. 89-95.
- [Lambert] Lambert J. D., *Computational Methods in Ordinary Differential Equations*, John Wiley Editors, New York, 1979.
- [López] López J., Comunicación Personal, México, D. F. UNAM, Facultad de Ciencias. 2000
- [López_1] López J. L., *Diseño, Construcción e Implantación de un Código para la Solución de Discontinuidades y Rigidez en Ecuaciones Diferenciales, Bajo Simulación en Tiempo Real*, Tesis de Licenciatura, Durango, Mex. UAD, 1992.
- [Luyben] Luyben W. L., *Process Modeling, Simulation and Control for Chemical Engineers*, McGraw-Hill Inc., 1981.
- [Ojeda_1] Ojeda R. B., *Ordenamiento de un Sistema de Ecuaciones Algebraicas no Lineales, para Reducir el Grado de Acoplamiento entre Ellas, Aplicado a Simulación Dinámica de Procesos*, Tesis de Licenciatura, México D.F., UNAM, 1986.

- [Ojeda_2] Ojeda R. B. et al, *Técnicas Numéricas Utilizadas en el Análisis Dinámico y Estructural de los Modelos del Simulador de Laguna Verde*, Boletín IIE, Marzo/Abril 1992, pp. 82-89.
- [Petzold_1] Petzold L. R., *A Description of DASSL a Differential/Algebraic System Solver*, Scientific Computing, 1983, pp. 65-68.
- [Petzold_2] Petzold L. R. et al, *Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems*.
- [Rodríguez_1] Rodríguez G. G., *Análisis de Técnicas Numéricas Para Tiempo Real*, Reporte Interno Depto. Simulación IIE, Cuernavaca Mor., Julio 1995.
- [Rodríguez_2] Rodríguez G. G., *Aproximación Eficiente de Jacobianos Para Sistemas No Lineales Aplicado a Simulación*, Proc. Del IV Coloquio de Control Automático, México, D.F., 1985.
- [Rodríguez_3] Rodríguez G. G., *Capacidad de Procesamiento Numérico de la Estación de Trabajo Sparc 10*, Reporte Interno Depto. Simulación IIE, Cuernavaca Mor., Septiembre 1995.
- [Rodríguez_4] Rodríguez G. G., *Implantación y Validación del Método Resolvedor de Sistemas de Ecuaciones Diferenciales Algebraicas a Paso Fijo Para Simulación en Tiempo Real*, Reporte Interno Depto. Simulación IIE, Cuernavaca Mor., Septiembre 1995
- [Rodríguez_5] Rodríguez G. G. et al, *Problemas en Aritmética de Punto Flotante y Obtención de Características Internas de Computadoras Digitales*, Boletín del IIE, Vol. 6, No. 1, Enero 1982, pp. 24-28.
- [Roswitha] Roswitha M. *Numerical Methods for Differential-Algebraic Equations Part II: Numerical Integration Methods*, Humboldt-Universität zu Berlin.
- [Rudin] Rudin W., *Principles of Mathematical Analysis*, McGraw-Hill Kogakusha, Third Edition, 1976.
- [Shampine_1] Shampine L. F., *Computer Solution of Ordinary Differential Equations*, W. H. Freeman and Company, 1975.
- [Shampine_2] Shampine L. F., *Implementation of Implicit Formulas for the Solution of ODEs*, SIAM J. Sci. Stat. Comput., Vol. 1, No. 1, Marzo 1980.
- [Usoro] Usoro P. B., *Modeling and Simulation of a Drum Boiler-Turbine Power Plant Under Emergency State Control*, M. S. Thesis, Department of Mechanical Engineering, MIT, 1997.