



INIS-MX--264



MX0000017

INSTITUTO POLITECNICO NACIONAL

**ESCUELA SUPERIOR DE FISICA Y MATEMATICAS
DEPARTAMENTO DE MATEMATICAS**

**GENERACION DE BANCOS DE DATOS
NUCLEARES MEDIANTE
INTERPOLACION**

T E S I S

P R E S E N T A D A

**ANTE EL COLEGIO DE PROFESORES DE LA SECCION DE
GRADUADOS DE LA ESCUELA SUPERIOR DE FISICA Y
MATEMATICAS DEL INSTITUTO POLITECNICO NACIONAL, PARA
SATISFACER, EN PARTE, LOS REQUISITOS NECESARIOS PARA
LA OBTENCION DEL GRADO DE**

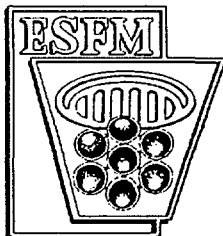
**MAESTRO EN CIENCIAS
(MATEMATICAS)**

POR

JOSE ALEJANDRO CASTILLO MENDEZ

DIRIGIDA POR:

Dr. ISIDRO ROMERO MEDINA



MEXICO, D. F.

Julio de 1999

31 - 18



SECRETARIA DE EDUCACION PUBLICA

INSTITUTO POLITECNICO NACIONAL

DIRECCION DE ESTUDIOS DE POSGRADO E INVESTIGACION

DIVISION DE ESTUDIOS DE POSGRADO

ACTA DE REVISION DE TESIS

8190516
Número de registro

En la ciudad de Mexico, D.F., siendo las 13:00 horas del día 18 del mes de mayo de 1999, se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de la E.S.F.M., para examinar la tesis de grado titulada:

" OBTENCION DE BANCOS DE DATOS NUCLEARES MEDIANTE INTERPOLACION "

presentada por el alumno:

LIC. JOSE ALEJANDRO CASTILLO MENDEZ

aspirante al grado de:

MAESTRO EN CIENCIAS ESPECIALIDAD EN MATEMATICAS

Después de intercambiar opiniones los miembros de la Comisión manifestaron SU APROBACION DE LA TESIS, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISION REVISORA

DR. ISIDRO ROMERO MEDINA (Director de tesis)

DR. GERMAN GONZALEZ SANTOS

DR. ROBERTO S. ACOSTA ABREU

DR. EDMUNDO DEL VALLE CALLEGOS

DR. JULIAN TABIA AGUILAR

EL PRESIDENTE DEL COLEGIO

DR. JESUS PALACIOS GOMEZ

Ya no vivo, pero voy en lo que andaba soñando.
Mi tumba no anden buscando porque no la encontrarán,
Mis manos son las que van, en otras manos tirando.
No me pregunten la edad, tengo los años de todos,
Yo elegí entre muchos modos ser más viejo que mi edad.
Y mis años de verdad son los tiros que he tirado,
nazco en cada fusilado y aunque el cuerpo se muera,
tendrá la edad verdadera del niño que he liberado.
Y sepan que solo muero, si ustedes van aflojando,
porque el que murió peleando vive en cada compañero.

Anónimo salvadoreño

A mis padres

Con todo respeto y cariño. Pues con sus manos trabajadoras me brindaron la oportunidad de saber un poco más. Con su ejemplo de honestidad y trabajo, formaron parte de lo que ahora soy y con su posición ante las situaciones más difíciles, me enseñaron lo importante que es luchar, buscar lo justo y el bien común, a pesar de todo.

A mis hermanos

Manuel, Toño, Paty y Bety.

A mi compañera de toda la vida, Raquel

Con quién aprendí lo difícil que es construir algo en la vida y sentir la satisfacción de verlo concluido. Al iniciar este trabajo estaba a su lado, hoy me siento muy feliz de que al concluirlo, sigamos juntos. Con todo mi amor.

A mis hijos, Quitzé y Tonalli

A quienes les quite parte del tiempo que les corresponde para concluir este proyecto y lejos de reclamar, siempre me respondieron con una sonrisa que me impulsó a terminar.

A Paco Ornelas.

Con respeto y admiración.

Agradezco muy especialmente a los profesores Isidro Romero, Germán González, Edmundo del Valle, Julián Tapia, Roberto Acosta y Carlos Arredondo, los comentarios que hicieron, pues en mucho ayudaron a enriquecer este trabajo.

Agradezco muy afectuosamente a mis amigos: Martín Torres, por los comentarios y el apoyo en la parte matemática; Héctor Hernández por la contribución que hizo en la parte nuclear; Carlos Guillén, por su ayuda al momento de darle formato a todo el documento, incluyendo la presentación. A los tres, millón de gracias.

Igualmente, agradezco al Instituto Nacional de Investigaciones Nucleares por las facilidades que me proporcionaron durante el desarrollo del proyecto.

Doy las gracias a todos los compañeros del Departamento de Sistemas Nucleares por el apoyo que siempre he obtenido de su parte.

Contenido

Lista de Figuras	iii
Lista de Tablas	vi
Resumen	vii
Abstract	viii
Introducción	ix
1.- Bancos de Datos Nucleares	1
1.1.- Definición	1
1.2.- Proceso de Generación	3
1.3.- Estructura de un Banco	5
1.4.- Parámetros de un Banco	10
2.- Interpolación de Lagrange	12
2.1.- Ideas Básicas	13
2.2.- Observaciones	16
3.- Comportamiento de los Parámetros	18
3.1.- Parámetros y Variables	19
3.2.- Comportamiento de las Gráficas	27
4.- Interpolación Polinomial Bicúbica	28
4.1.- Interpolación con Elemento Finito	29
4.2.- Aplicación de la Técnica a un Banco	36
4.3.- Solución Empleando Factorización LU	41
4.3.1.- Número de Condición y Factorización LU	44
4.4.- Solución Empleando Diferencias Divididas	46
4.4.1.- Número de Condición y Diferencias Divididas	53

5.- Diseño del Sistema Intpolbi	56
5.1.- Descripción General	57
5.2.- Estructura del Módulo Interpol 1ra. Versión	58
5.3.- Modificaciones para la 2da. Versión	65
5.4.- Estructura del Módulo Evalua	67
5.5.- Conclusiones	71
6.- Resultados	73
6.1.- Comparación para k_{inf}	75
6.2.- Resultados para otros Parámetros	89
6.3.- Tiempos de CPU	93
6.4.- Número de Condición para IINTPOLBI	94
6.5.- Conclusiones	95
Apéndice A	101
A.1.- Requisitos	102
A.2.- Módulo de Interpolación	102
A.2.1.- Archivo de Entrada	106
A.2.2.- Errores del Módulo	113
A.3.- Módulo de Evaluación	115
A.3.1.- Errores del Módulo	122
Referencias	124
Bibliografía	126

Lista de Figuras

1.1	Diagrama de un ensamble	2
1.2	Proceso para la generación de Bancos de Datos Nucleares	4
1.3	Relación entre cada bloque de un Banco Nuclear	6
2.1	Comportamiento con el polinomio de Lagrange	15
2.2	Interpolación de Lagrange tomando diferentes puntos	16
3.1	Comportamiento de Σ_{f2} con quemado 0	21
3.2	Comportamiento de Σ_{f2} con quemado 10	21
3.3	Comportamiento de Σ_{f2} con quemado 15	22
3.4	Comportamiento de $\nu\Sigma_{f2}$ con quemado 0	22
3.5	Comportamiento de $\nu\Sigma_{f2}$ con quemado 10	23
3.6	Comportamiento de $\nu\Sigma_{f2}$ con quemado 20	23
3.7	Comportamiento de Σ_{a2} con quemado 0	24
3.8	Comportamiento de Σ_{a2} con quemado 10	24
3.9	Comportamiento de Σ_{a2} con quemado 15	25
3.10	Comportamiento de D_2 con quemado 10	25
3.11	Comportamiento de D_2 con quemado 15	26
3.12	Comportamiento de D_2 con quemado 20	26
4.1	Una discretización de elemento finito	30
4.2	Relación entre el número de términos del interpolante, los elementos y sus nodos	31
4.3	Frontera de dos elementos contiguos	34
4.4	Producto tensorial para elementos rectangulares	38
4.5	Marco de referencia para el sistema obtenido con elemento finito	40
4.6	Discretización del dominio, vista en el plano XY	47
5.1	Estructura del sistema INTPOLBI	58
5.2	Estructura del módulo INTERPOLA	59
5.3	Flujo de información del módulo INTERPOLA	60
5.4	Estructura del módulo EVALUA	68
5.5	Flujo de información del módulo EVALUA	69

6.1	Errores para k_{infr} caso 2, Bancos L1REFC*1.REC	79
6.2	Errores para k_{infr} caso 5, Bancos L1REFC*2.REC	80
6.3	Errores para k_{infr} caso 1, Bancos L1REFC*3.REC	81
6.4	Errores para k_{infr} caso 4, Bancos L1REFC*4.REC	82
6.5	Errores para k_{infr} caso 1, Bancos L1REFC*5.REC	83
6.6	Errores para k_{infr} caso 2, Bancos L1REFC*6.REC	84
6.7	Errores para k_{infr} caso 5, Bancos L1REFC*7.REC	85
6.8	Errores para k_{infr} caso 4, Bancos L1REFC*8.REC	86
6.9	Errores para k_{infr} caso 1, Bancos L1REFC*9.REC	87
6.10	Errores para k_{infr} caso 4, Bancos L1REFC*10.REC	88
A.1	"Banner" del módulo de interpolación	103
A.2	Captura del archivo de entrada	104
A.3	Mensaje de error en la apertura del archivo de entrada	104
A.4	Mensaje de error en la lectura del archivo de entrada	105
A.5	Mensaje de error en el número de Bancos proporcionado	105
A.6	Mensaje de error en el número de Bancos proporcionado	105
A.7	Lectura del nombre del Banco de salida	105
A.8	Lectura del nombre del archivo de coeficientes	106
A.9	Archivo de entrada	108
A.10	Lectura del porcentaje de gadolinia	108
A.11	Error en el porcentaje de gadolinia	109
A.12	Lectura del porcentaje de uranio	109
A.13	Error en el porcentaje de uranio	109
A.14	Mensaje inicial para desplegar datos de entrada	110
A.15	Despliegue de los Bancos de entrada	110
A.16	Despliegue de los identificadores de las celdas	111
A.17	Despliegue de los parámetros de entrada	111
A.18	Información durante el proceso	112
A.19	Mensaje si la respuesta es negativa	112
A.20	Mensaje para indicar el fin de la ejecución	113
A.21	Mensaje de error	113
A.22	Mensaje de error	113
A.23	Mensaje de error	114
A.24	Mensaje de error	114
A.25	Mensaje de error	114
A.26	Mensaje de error	115
A.27	"Banner" del módulo de evaluación	116
A.28	Captura del archivo de entrada	116
A.29	Lectura del archivo de coeficientes	117
A.30	Mensaje de error en la apertura del archivo de coeficientes	117
A.31	Lectura del Banco de entrada	117
A.32	Mensaje de error en la apertura del Banco de entrada	118

A.33	Mensaje de error si el Banco de entrada no es el adecuado	118
A.34	Lectura del nombre del Banco de salida	118
A.35	Mensaje inicial para desplegar datos de entrada	119
A.36	Despliegue de los Bancos utilizados en la interpolación	119
A.37	Despliegue de los identificadores de las celdas	120
A.38	Despliegue de los parámetros de entrada	120
A.39	Información durante el proceso	121
A.40	Mensaje si respuesta es negativa	121
A.41	Mensaje para indicar el fin de la ejecución	122
A.42	Mensaje de error	122
A.43	Mensaje de error	123
A.44	Mensaje de error	123
A.45	Mensaje de error	123

Lista de Tablas

1.1	Valores para T	7
1.2	Estructura de un "caso"	9
3.1	Porcentajes de ^{235}U y Gd_2O_3	20
4.1	Producto tensorial para obtener los polinomios en el caso de rectángulos	37
6.1	Porcentajes de Gd_2O_3 y ^{235}U empleados para generar la malla	75
6.2	Porcentajes de Gd_2O_3 y ^{235}U de los Bancos obtenidos con INTPOLBI (ambas versiones) durante las pruebas	76
6.3	Porcentajes de Gd_2O_3 y ^{235}U de los Bancos obtenidos con INTERTEG	77
6.4	Porcentajes de Gd_2O_3 y ^{235}U de los Bancos obtenidos con INTERTEG	78
6.5	Errores para k_{inf} caso 2, Bancos L1REFCI1 y L1REFCJ1	80
6.6	Errores para k_{inf} caso 5, Bancos L1REFCI2 y L1REFCJ2	81
6.7	Errores para k_{inf} caso 1, Bancos L1REFCI3 y L1REFCJ3	82
6.8	Errores para k_{inf} caso 4, Bancos L1REFCI4 y L1REFCJ4	83
6.9	Errores para k_{inf} caso 1, Bancos L1REFCI5 y L1REFCJ5	84
6.10	Errores para k_{inf} caso 2, Bancos L1REFCI6 y L1REFCJ6	85
6.11	Errores para k_{inf} caso 5, Bancos L1REFCI7 y L1REFCJ7	86
6.12	Errores para k_{inf} caso 4, Bancos L1REFCI8 y L1REFCJ8	87
6.13	Errores para k_{inf} caso 1, Bancos L1REFCI9 y L1REFCJ9	88
6.14	Errores para k_{inf} caso 4, Bancos L1REFCI10 y L1REFCJ10	89
6.15	Resultados globales para los códigos analizados	91
6.16	Resultados globales para los códigos analizados	92
6.17	Tiempos de CPU consumidos por los códigos	94
6.18	Números de condición para las dos versiones de INTPOLBI	95

Resumen

La generación de Bancos de Datos Nucleares es un proceso en el cual se invierte una buena cantidad de recursos, tanto de cómputo como humanos. Si se toma en cuenta que en ocasiones es necesario generar una buena cantidad de éstos, conviene contar con una herramienta confiable que genere dichos Bancos con la menor cantidad de recursos, en el menor tiempo posible y con una muy buena aproximación.

En este trabajo se presentan los resultados obtenidos al desarrollar el código INTPOLBI, el cual genera Bancos de Datos Nucleares empleando interpolación polinomial bicúbica, tomando como variables independientes los porcentajes de uranio y gadolinia. Se trabajaron dos propuestas, aplicando en ambos casos el método de elemento finito, utilizando un elemento con 16 nodos para llevar a cabo la interpolación. En la primer propuesta se empleó la base canónica para obtener el polinomio interpolante primero y posteriormente, el sistema de ecuaciones lineales correspondiente. En la solución de éste último se aplicó el método de eliminación Gaussiana con pivoteo parcial. En el segundo caso, se utilizó la base de Newton para obtener el sistema mencionado, resultando una matriz triangular inferior, cuya estructura se aprovechó, aplicándole operaciones elementales, para obtener una matriz diagonal por bloques con características muy especiales y fácil de trabajar.

Para las pruebas de validación, se hizo la comparación entre los valores obtenidos con los códigos INTPOLBI e INTERTEG (este fue creado en el Instituto de Investigaciones Eléctricas con el mismo fin) y Bancos generados con el proceso convencional, esto es, con los códigos nucleares que habitualmente se trabaja.

Finalmente, los resultados obtenidos con el código INTPOLBI permiten afirmar que los Bancos de Datos Nucleares generados con éste, son una muy buena aproximación, que si bien no sustituyen por completo el proceso convencional, si ayudan en muchos casos cuando es necesario generar una gran cantidad de Bancos.

Abstract

Nuclear Data Bank generation, is a process in which a great amount of resources is required, both computing and humans. If it is taken into account that at some times it is necessary to create a great amount of those, it is convenient to have a reliable tool that generates Data Banks with the lesser resources, in the least possible time and with a very good approximation.

In this work are shown the results obtained during the development of INTPOLBI code, used to generate Nuclear Data Banks employing bicubic polynomial interpolation, taking as independent variables the uranium and gadolinia percents. Two proposals were worked, applying in both cases the finite element method, using one element with 16 nodes to carry out the interpolation. In the first proposals the canonic base was employed, to obtain the interpolating polynomial and later, the corresponding linear equations system. In the solution of this system the Gaussian elimination method with partial pivot was applied. In the second case, the Newton base was used to obtain the mentioned system, resulting in a triangular inferior matrix, which structure, applying elemental operations, to obtain a blocks diagonal matrix, with special characteristics and easier to work with.

For the validations tests, a comparison was made between the values obtained with INTPOLBI and INTERTEG (created at the Instituto de Investigaciones Eléctricas with the same purpose) codes, and Data Banks created through the conventional process, that is, with nuclear codes normally used.

Finally, it is possible to conclude that the Nuclear Data Banks generated with INTPOLBI code constitute a very good approximation that, even though do not wholly replace conventional process, however are helpful in cases when it is necessary to create a great amount of Data Banks.

Introducción

Como parte de las actividades que desarrolla el grupo de administración de combustible nuclear del Instituto Nacional de Investigaciones Nucleares (ININ), se encuentra la generación de Bancos de Datos Nucleares, base principal para la simulación de los reactores nucleares en operación de la Central Nucleoeléctrica de Laguna Verde (CNLV). Estos Bancos de Datos contienen información representativa (secciones eficaces macroscópicas ponderadas para dos grupos de energía, pico de potencia local, k -infinita, entre otras) de una sección de un ensamble de combustible nuclear con determinadas características, tales como: enriquecimiento de uranio (contenido en peso de ^{235}U), número de barras y contenido de "veneno" quemable como lo es la gadolinia (Gd_2O_3); a dicha sección se le denomina celda combustible.

El proceso para generar un Banco de Datos de una celda combustible consume una gran cantidad de recursos, entre ellos tiempo de cómputo, por lo que al diseñar un ensamble de combustible en el que se utilicen varias celdas combustibles eleva considerablemente el consumo de dichos recursos, y por ende, el proceso se vuelve lento, además de que se requiere de una base de datos muy grande, lo cual conlleva en ocasiones a la saturación del sistema de cómputo en el cual se trabaja. En este sentido, se planteó la necesidad de tener una herramienta capaz de producir confiablemente un Banco de Datos empleando la menor cantidad de

recursos posible, a partir de una base de datos más pequeña ya existente, para resolver esta problemática se pensó en utilizar un método de interpolación con dos variables independientes como lo son el enriquecimiento de uranio (^{235}U) y el contenido de veneno quemable, esto es la gadolinia (Gd_2O_3).

El presente trabajo pretende cubrir algunos aspectos importantes: primero, explicar de manera breve y sencilla las ideas fundamentales para la generación de un Banco de Datos Nucleares; segundo, dejar claro en cual de los pasos necesarios para la generación de Bancos se pudo incidir; tercero, cubrir los aspectos técnicos de manera más profunda y formal en lo que se refiere al proceso de interpolación utilizada; por último, presentar los resultados obtenidos hasta el momento con el sistema de cómputo creado para tal fin (*INTPOLBI*).

Por otro lado, es importante señalar que, una vez que se concluyó el sistema *INTPOLBI*, se continuó el trabajo buscando una segunda versión de éste, tratando de encontrar mejores resultados al modificar algunos métodos empleados, en ese sentido, se logró mejorar la primera versión del sistema cambiando algunos procesos por otros más eficaces, dicha propuesta se incluye en el presente trabajo, así como los resultados obtenidos.

El trabajo contiene 6 capítulos y un apéndice, desglosados de la manera siguiente: el capítulo 1 sirve para describir el proceso completo para la generación de un Banco de Datos Nucleares, haciendo hincapié en cual de los pasos intermedios se trabajó, asimismo se analiza su estructura y los parámetros involucrados, remarcando aquellos que son considerados los más importantes por su repercusión en el Banco. En el capítulo siguiente se exponen algunas ideas que dieron origen a la interpolación, tomando como base el trabajo realizado en el Instituto de Investigaciones Eléctricas (IIE). En el capítulo 3 se describe el comportamiento de los valores de algunos parámetros involucrados en un Banco, al utilizar el método de interpolación que viene implementado en *MATLAB*, estos resultados fueron la antesala para finalmente contemplar la interpolación utilizada en el sistema *INTPOLBI*. En el capítulo 4 se explica el fundamento matemático de la interpolación polinomial bicúbica, haciendo hincapié en la metodología utilizada para resolver dicho problema e incluyendo las dos propuestas que se trabajaron. El capítulo siguiente se utilizó para presentar el diseño del sistema de cómputo creado, incluyendo los dos caminos que se siguieron para obtener las dos versiones del código *INTPOLBI*. El último capítulo está dedicado a la presentación de los resultados obtenidos con las dos versiones del método de interpolación, incluyendo además, una comparación entre las dos versiones del código *INTPOLBI* y el código propuesto en el IIE. Finalmente, en el apéndice se presenta el manual de usuario del sistema *INTPOLBI*, para en caso de así necesitarlo, el usuario este enterado de la forma de proporcionar los datos de entrada y al mismo tiempo, tenga conocimiento de los archivos de salida que se obtienen, así como su contenido.

Se puede afirmar que los dos capítulos más importantes dentro del presente trabajo son, por una parte el que habla de la estructura de los Bancos de Datos Nucleares (Cap. 1), ya que es la base fundamental para entender todo el proceso que se llevará a cabo y, en segundo término, el correspondiente a la interpolación polinomial bicúbica (Cap. 4), pues es la piedra angular del trabajo realizado.

Es necesario hacer notar que el presente trabajo concluye finalmente en el sistema de cómputo creado (*INTPOLBI*) para obtener los Bancos de Datos Nucleares, por lo cual se debe hacer hincapié en la naturaleza misma del sistema.

Si bien es cierto el sistema *INTPOLBI* no puede, por el momento, señalarse como una herramienta que sustituya la metodología actual para la generación de los Bancos, sí podemos afirmar que es una ayuda sumamente importante para la generación de éstos, debido a los resultados obtenidos, la facilidad de manejo, los bajos costos que implica su buen funcionamiento y a la reducción en el tiempo requerido para obtener un nuevo Banco de Datos Nucleares. No obstante, se sigue trabajando en nuevas metodologías, que junto con las ya existentes, reduzcan de manera considerable el tiempo para la obtención de dichos Bancos.

Finalmente, es importante señalar que el trabajo realizado hasta el momento para la obtención del sistema *INTPOLBI* se llevó a cabo de manera interdisciplinaria, relacionando en todo momento las Matemáticas, la Ingeniería Nuclear y la Computación.

Capítulo 1

Bancos de Datos Nucleares

Como se sabe, el primer paso para el desarrollo de un sistema de cómputo, es la comprensión adecuada de todo aquello con lo que se va a trabajar, incluyendo las variables más importantes. En este caso, la materia prima fundamental son los Bancos de Datos Nucleares, en este sentido trataremos de explicar, aunque sea de manera breve, su definición y el proceso para la generación de éstos. Particularmente dentro de los pasos que se siguen para la generación de dichos Bancos, se hará énfasis en que parte del proceso se trabajo de manera especial y por último la estructura básica de un Banco de Datos Nucleares. Asimismo, se expondrán los conceptos y variables más importantes involucradas, obviamente no se trata de una explicación profunda, puesto que el fin del presente trabajo es otro.

1.1.- Definición

Sin entrar en muchos detalles, se puede hacer una definición de un Banco de Datos Nucleares a partir de uno de los componentes principales del núcleo de un reactor, esto es, un ensamble de combustible nuclear. Un ensamble de combustible está formado principalmente por un conjunto de barras combustibles en un arreglo

geométrico definido, el número de barras varía dependiendo del tipo de ensamble. A su vez, un ensamble se divide axialmente en secciones de acuerdo al porcentaje en el contenido de ^{235}U y Gd_2O_3 , material utilizado para su creación, separación entre barras, etc. Cada una de estas secciones se conoce con el nombre de *celda* y a partir de esta "unidad básica" se forma un Banco de Datos Nucleares [1]. En la Figura 1.1 se muestra un diagrama sencillo en dirección axial de la región con combustible de un ensamble, especificando la longitud que en general tienen y al mismo tiempo señalando la sección correspondiente a las celdas en que se divide.

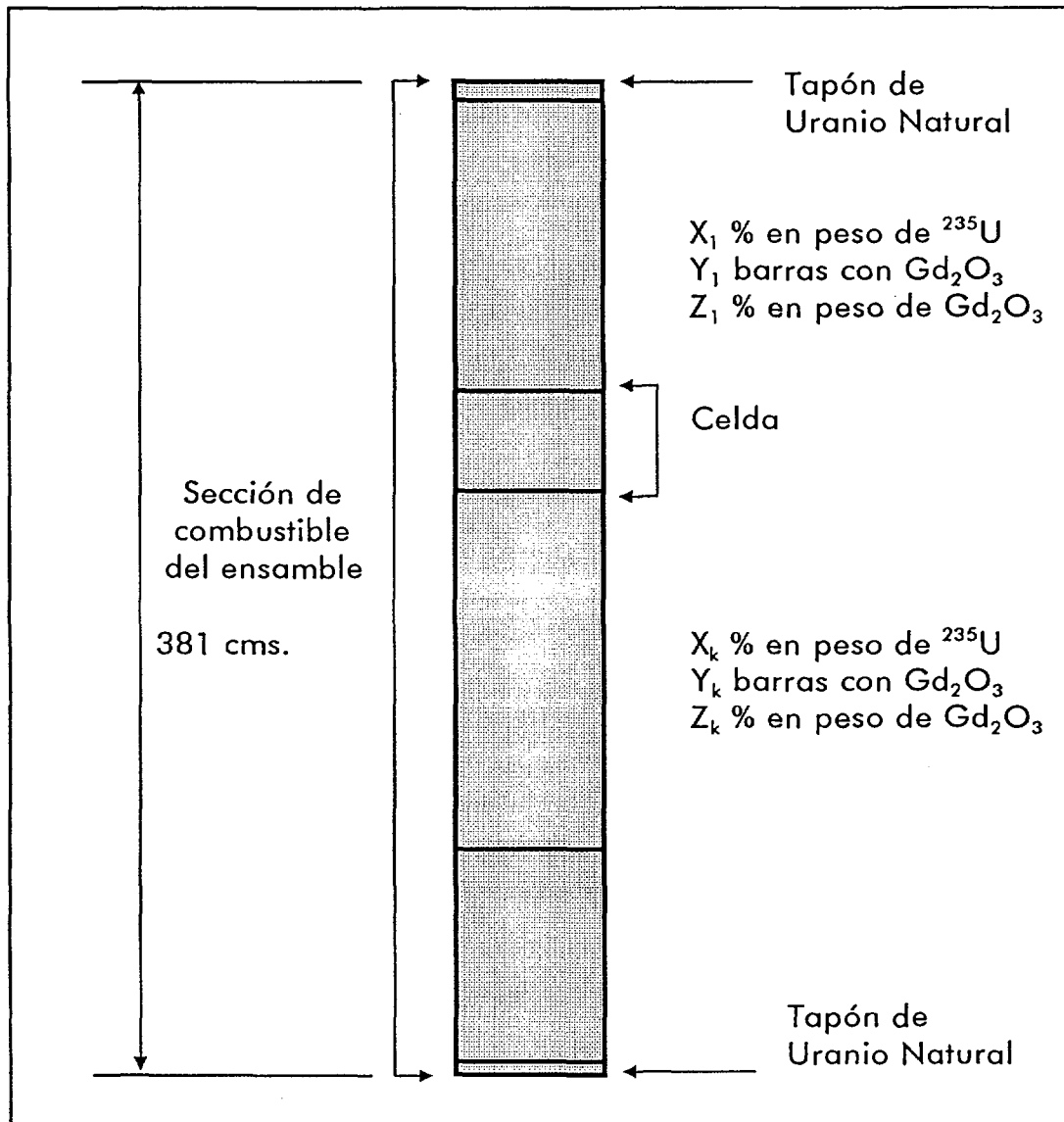


Figura 1.1. Diagrama de un ensamble.

Un Banco de Datos Nucleares es en términos generales, un archivo que contiene un conjunto de parámetros físicos (secciones eficaces transversales macroscópicas, coeficientes de difusión en dos grupos de energía, k-infinita y pico de potencia local), representativos de la celda, calculados en diferentes condiciones físicas predeterminadas, con la finalidad de poder simular la operación real de un reactor, tomando en cuenta las condiciones a las que es operado en función del quemado del combustible¹.

Con la idea fundamental de simular lo más cercano a la realidad el funcionamiento de un reactor, se requiere determinar los parámetros físicos representativos de la configuración de cada celda en particular; esto constituye en esencia, el proceso de generación de Bancos.

1.2.- Proceso de Generación

A continuación se hará una descripción general del proceso de generación de un Banco de Datos Nucleares, con la idea de señalar de forma especial en que parte del proceso se trabajó. Para entender más dicho proceso, en la Figura 1.2 se presenta un diagrama del proceso mencionado.

Paso 1) Se proporcionan al código *RECORD* los datos geométricos, la distribución de materiales, las condiciones de operación, etc., de una celda combustible.

Paso 2) El código *RECORD* se encarga de obtener un Banco de Datos Nucleares de la celda descrita anteriormente. Este contiene las secciones eficaces macroscópicas y otros parámetros de la celda, tabulados en función del quemado.

Paso 3) Dadas las implicaciones que resultan al trabajar con datos tabulados, las tablas del Banco de Datos se transforman en polinomios (también en función del quemado), por medio del código *POLGEN*, es decir:

$$\begin{array}{c} \text{POLGEN} \\ \Sigma = F(Q) \Rightarrow \Sigma_x = A_0 + A_1Q + A_2Q^2 + \dots + A_nQ^n \end{array} \quad 1.1$$

¹ Se define como la energía total liberada por unidad de masa del material combustible cargada adentro del reactor. Comúnmente se expresa en unidades de megawatt días (MWD)/kilogramo de combustible o MWD/ton. métrica de combustible.

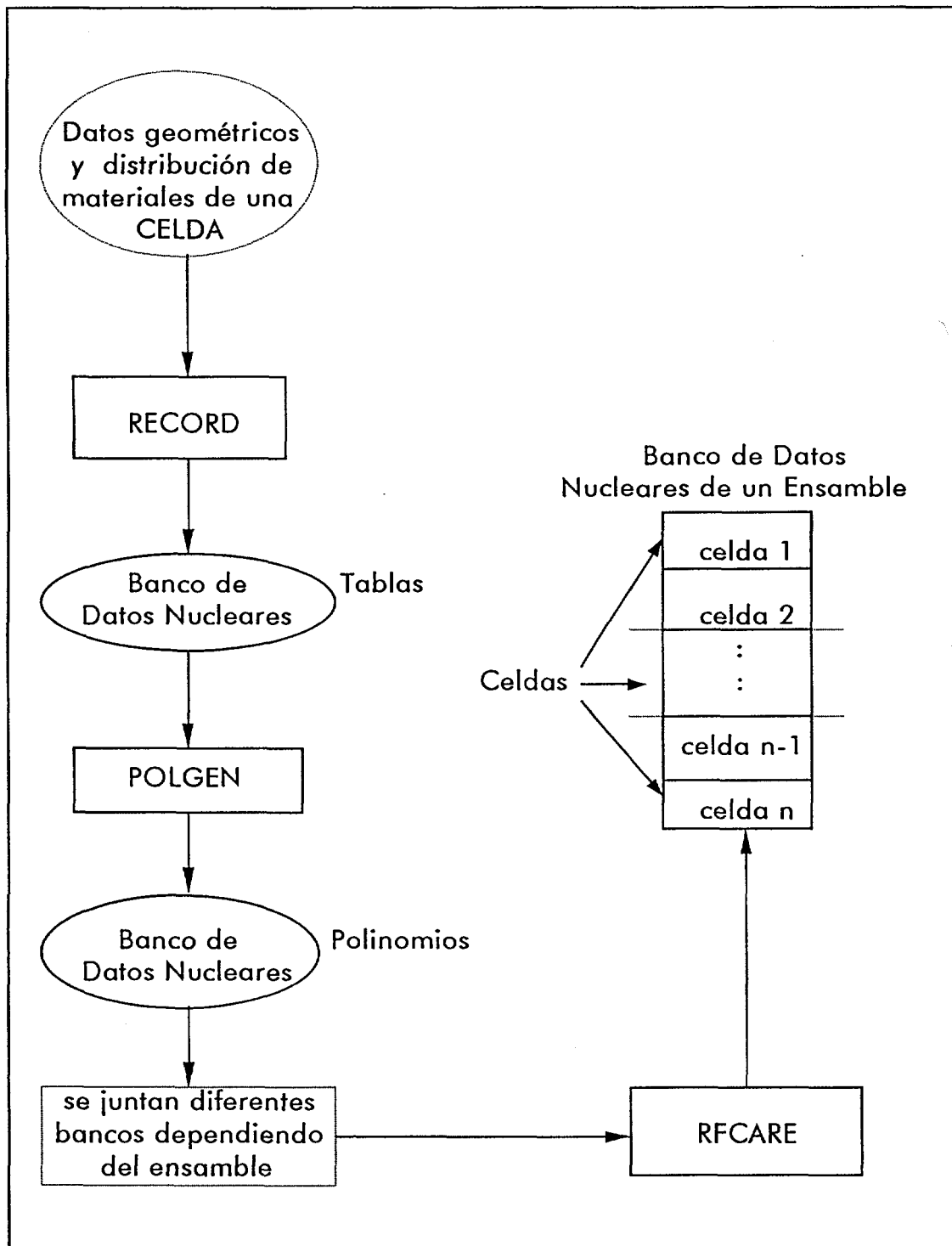


Figura 1.2. Proceso para la generación de Bancos de Datos Nucleares.

donde

Σ : es la sección eficaz macroscópica

Σ_x : puede ser la sección de absorción, de fisión, de dispersión, etc.

Q : es el quemado

Los pasos 1, 2 y 3 se repiten tantas veces como el número de celdas identificadas existan, en un ensamble de combustible.

Paso 4) Una vez que se tienen los Bancos de Datos para cada una de las celdas de un ensamble de combustible, se unen al de otros ensambles a través del código *RFCARE*, con esto se obtiene un Banco de Datos que contiene los parámetros antes descritos, para cada uno de los diferentes tipos de ensambles de combustible que se tienen dentro del núcleo del reactor.

En esencia, los pasos anteriores describen el proceso llevado a cabo para la generación de un Banco de Datos Nucleares; a simple vista el proceso resulta bastante sencillo. Sin embargo, en el paso 2 se consume una gran cantidad de tiempo de CPU y de recursos del sistema, además de que se vuelve dependiente del paso 1. En este sentido, el grupo de administración de combustible del ININ, requería de una herramienta capaz de generar una gran base de datos de Bancos Nucleares, que aumentara su capacidad de predicción en el menor tiempo posible y que se formara a partir de los Bancos de Datos Nucleares ya existentes. De esta forma se identificó que esta herramienta se podría desarrollar a partir de los Bancos tabulados empleando la interpolación polinomial bicúbica.

1.3.- Estructura de un Banco

Un Banco Nuclear está estructurado en forma de bloques, en principio se pueden hacer dos grandes divisiones perfectamente definidas, el primer bloque proporciona información acerca del segundo, éste a su vez se divide en varias partes conocidas como "*casos*". A partir de las ideas anteriores, se tratará de explicar tanto la relación existente entre los bloques de información, como el contenido de cada una de las partes del segundo bloque.

Cada "*caso*" del segundo bloque contiene las diferentes condiciones o estados de operación del reactor²; por otra parte, existe un "*caso*" por cada línea del primer

² Se refiere a condiciones tales como, temperatura de operación, con presencia de barra de control o sin ella, con Xenón o sin él, etc.

bloque, en dicha línea se coloca el identificador del "caso", es decir, en el primer bloque de información tengo tantas líneas como "casos" en el segundo bloque, en la Figura 1.3 se muestra la correspondencia entre ambos bloques.

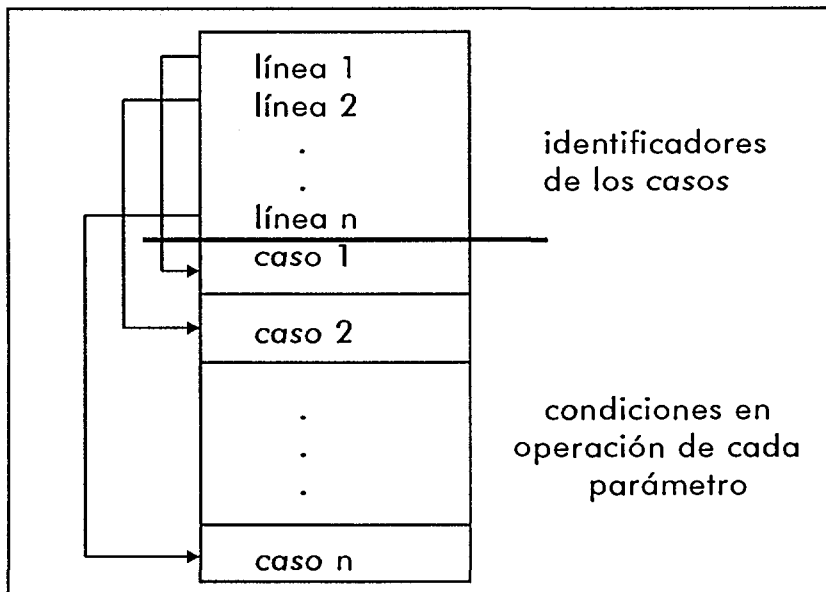


Figura 1.3. Relación entre cada bloque de un Banco Nuclear.

Continuando con el contenido del primer bloque tenemos que el identificador mencionado, proporciona la clasificación de cada uno de los "casos" (contenidos en el segundo bloque) que tiene la celda, dicha información consta de 500 datos, que se pueden ver como grupos de 5⁽³⁾, cada uno con el identificador de cada "caso", es decir, a lo más podemos tener 100 "casos" en un Banco. La información de cada grupo se desglosa de la manera siguiente: el primer dato es un valor numérico en formato real que proporciona el número de "caso" en orden ascendente e iniciando con 1.0; el segundo y tercer datos son cadenas de 10 caracteres cada una, que proporcionan los identificadores de la celda para cada "caso", de acuerdo a la siguiente nomenclatura:

Para el primer identificador (2do dato)

bbbbREbSQ

³ Cada grupo de 5 datos es una línea.

donde

- b* : representan espacios en blanco
- RE : cadena fija, en ocasiones se acostumbra poner RC
- SQ : secuencia de estudio (2 caracteres)

Para el segundo identificador (3er dato)

bbbbCCTbXX

donde

- b* : representan espacios en blanco
- CC : número de celda analizada
- T : tipo de cálculo
- XX : condiciones del tipo de cálculo {00,01,02,03,...}

los valores de T se obtienen de acuerdo a la Tabla 1.1.

T	Tipo de cálculo
1	A potencia total sin barra de control
2	A potencia total con barra de control
3	Efecto Doppler
4	Efecto del Xenón
5	Historia de vacíos (BRANCH_OFF)

Tabla 1.1. Valores para T.

Finalmente, los datos cuarto y quinto son valores reales que se utilizan para saber el número total de parámetros que tiene cada "caso", lo anterior con base en la fórmula siguiente:

$$NTPC = D4 + D5*20$$

1.2

donde

NTPC : Número total de valores para el "caso" analizado

D4 : Cuarto dato

D5 : Quinto dato

Observación :

si $D4 < 20$ entonces $D4 = 20$

si $D5 < 28$ entonces $D5 = 28$

Es importante hacer notar que aunque existan menos de cien "casos" en un Banco, siempre deben aparecer los 500 datos, tomando en cuenta que una vez agotados los "casos", los datos siguientes son todos cero, si son numéricos, o espacios en blanco si son caracteres.

A continuación se analizará la segunda parte del Banco, esto es, los "casos"; cada uno de éstos está asignado a un tipo de estudio que se calcula con el código *RECORD*, se les conoce como "casos de la celda" y representan la simulación en operación de la celda a diferentes condiciones físicas.

Para hacer un mejor análisis de dicha información, bastará con describir la estructura de un "caso", con el entendido de que los restantes tienen exactamente la misma estructura. Cada "caso" se puede a su vez dividir en dos grupos, el primero representa la descripción de las condiciones de cálculo de los parámetros; por otra parte, el segundo grupo contiene, en ese orden: las concentraciones isotópicas dentro de la celda, los parámetros nucleares macroscópicos y por último los parámetros nucleares dinámicos [2]; el segundo grupo al que se hace mención se agrupa en conjuntos de 20 elementos, cada uno representa la simulación de la celda con respecto al parámetro, a diferentes pasos de quemado, cuyo valor puede ser 0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 11000, 12500, 15000, 20000, 25000, 30000, 40000, 50000 y 60000 MWd/t.

Cada "caso" contiene 878 elementos, de los cuales, los primeros 38 de estos pertenecen al primer grupo descrito anteriormente; mientras que en el segundo grupo se encuentran los restantes. Por último, conviene mencionar que el archivo que contiene al Banco de Datos Nucleares es un archivo binario, del cual se conoce el formato.

La Tabla 1.2 muestra un resumen de lo anteriormente dicho, con lo cual damos por terminado este análisis, considerando que están sentadas las bases para una mejor comprensión de la estructura de un Banco.

Dato	Característica
1	Número de caso
2 y 3	Identificadores de la celda
4	Densidad de potencia
5	Enriquecimiento de uranio en ^{235}U
6	Porcentaje de vacíos
7	Area de combustible
8	Número de pasos de quemado
9 al 28	Pasos de quemado
29	Gramos de Uranio por centímetro cúbico
30	Número de pastillas con combustible
31	Densidad del moderador
32	Temperatura del combustible
33	Temperatura del moderador
34	Fracción del moderador
35 al 38	Registros no utilizados en la actualidad
39 al final	Aparecen en grupos de 20, cada uno de estos representa la simulación de la celda con respecto al parámetro, a diferentes pasos de quemado.

Tabla 1.2. Estructura de un "caso".

1.4.- Parámetros de un Banco

En la sección anterior se analizó la estructura que tiene un Banco de Datos Nucleares, particularmente se mencionó que cada "caso" contiene 878 elementos, si no se toman en cuenta los primeros 38 valores que se refieren a las condiciones de cálculo, se tiene que los últimos 840 pertenecen a los diferentes parámetros del Banco, como dichos valores se encuentran en grupos de 20, entonces existen 42 parámetros en total para cada "caso"⁴. De todos los parámetros involucrados 12 juegan un papel muy importante, pues sirven para decidir en buena medida si se generó adecuadamente un Banco, estos 12 parámetros se pueden dividir a su vez, en dos grupos de acuerdo al impacto que tienen en la generación de un Banco; el primer grupo consta de 11 elementos y el segundo de únicamente 1, a continuación se dan los nombres de dichos parámetros y al mismo tiempo para que se hizo dicha separación del total de parámetros involucrados.

El primer grupo al cual se hizo referencia involucra los parámetros mostrados a continuación, en la lista también se incluye la nomenclatura que se utiliza comúnmente para nombrarlos y la cual se adoptará a lo largo del documento.

- 1) Coeficiente de difusión del grupo 1 (D_1)
- 2) Coeficiente de difusión del grupo 2 (D_2)
- 3) Sección eficaz de absorción del grupo 1 (Σ_{a1})
- 4) Sección eficaz de absorción del grupo 2 (Σ_{a2})
- 5) Sección eficaz de remoción del grupo 1 al grupo 2 ($\Sigma_{R1,2}$)
- 6) Número de neutrones producidos por fisión en el grupo 1 ($\nu\Sigma_{f1}$)
- 7) Número de neutrones producidos por fisión en el grupo 2 ($\nu\Sigma_{f2}$)
- 8) Sección eficaz de fisión del grupo 1 (Σ_{f1})
- 9) Sección eficaz de fisión del grupo 2 (Σ_{f2})
- 10) Espectro de fisión del grupo 1 (X_1)
- 11) Espectro de fisión del grupo 2 (X_2)

Sin restarle importancia a los otros parámetros que se incluyen en un Banco, se puede afirmar que los 11 anteriores tienen el mayor impacto dentro de la simulación de un reactor en operación, debido a que son las secciones eficaces con las cuales se realiza la simulación y cálculo del reactor en operación.

Por otra parte, el segundo grupo mencionado incluye un solo elemento, dicho parámetro se conoce como k_{inf} (k infinita), cuya importancia reside en que sirve como una prueba inicial de validación bastante efectiva en la generación de un Banco, puesto que, representa la relación entre el número de neutrones que se

⁴ Se puede generalizar para todo el Banco, pues lo que cambia para cada "caso" son las condiciones de operación o estados, no los parámetros involucrados.

producen por fisión en cada generación y el número de neutrones absorbidos en la generación anterior, correspondiente a un sistema de tamaño infinito. En un sistema infinitamente grande no hay pérdidas de neutrones por escape, de modo que la desaparición de neutrones del sistema se produce solamente por absorción, en el combustible, moderador, etc. La condición de criticidad, es decir, aquella que hace justamente posible la fisión en cadena autosustentada, se reduce a que la k_{inf} sea igual a la unidad; en cuyo caso, el número de neutrones producidos en cada generación sería exactamente igual al número de neutrones perdidos en la generación anterior. De este modo, podría mantenerse un estado estacionario con densidad neutrónica constante, prosiguiendo la reacción en cadena con una velocidad constante y bien definida [3]. En otras palabras, el parámetro k_{inf} permite estimar la vida útil de la celda durante su permanencia en el reactor, además también se puede verificar que su funcionamiento se encuentre dentro de los límites de seguridad del mismo.

Asimismo, dicho valor esta en función de algunos de los parámetros mencionados anteriormente, como lo muestra la expresión siguiente

$$k_{inf} = \frac{\nu \sum f_1}{\sum a_1 + \sum R_1} + \frac{\sum R_1}{\sum a_1 + \sum R_1} \cdot \frac{\nu \sum f_2}{\sum a_2} . \quad 1.3$$

Conviene aclarar que el fin del presente trabajo no es dar una explicación profunda de los aspectos nucleares involucrados durante el proceso de generación de un Banco, por tal motivo, dejaremos hasta aquí la discusión anterior; simplemente se concluye que el análisis de dicho valor representa una prueba inicial adecuada para validar los resultados obtenidos en la generación de Bancos de Datos Nucleares.

Capítulo 2

Interpolación de Lagrange

La idea de encontrar una herramienta alternativa que cumpliera las características de exactitud, versatilidad, utilidad y bajo costo al generar un Banco de Datos Nucleares con respecto a la forma tradicional, surge de la gran cantidad de Bancos que se requieren para ciertos estudios del grupo de administración de combustible del ININ, relacionados con la Central Nucleoeléctrica de Laguna Verde (CNLV). Al iniciar la solución del problema se retomaron ciertas ideas utilizadas en el Instituto de Investigaciones Eléctricas (IIE), debido a que en dicha institución ya se había programado un código que generaba Bancos empleando la técnica de interpolación [4], en particular, el trabajo mencionado emplea los polinomios de Lagrange en una variable.

En el presente capítulo no se explicará de manera extensa la metodología en sí, únicamente algunos puntos importantes, pues el documento referido la explica de manera adecuada para los fines que persigue. Más bien se tratará de hacer un breve análisis sobre el por qué se considera que dicha metodología no es la más adecuada para aplicarse en la creación de Bancos Nucleares, salvo en ejemplos muy particulares.

2.1.- Ideas Básicas

Se sabe que el polinomio interpolante de Lagrange de grado n requiere de $n+1$ puntos en el plano para poder calcularse [5] (a diferencia de otros métodos que necesitan más información). Esta característica es la base principal del trabajo realizado en el IIE, pues en el caso de los Bancos Nucleares, fácilmente podemos conocer los valores a interpolar, que no son otra cosa que los valores de cada uno de los parámetros que lo componen¹, multiplicados por los 20 diferentes pasos de quemado. Sin embargo, en general tiene algunas desventajas dependiendo del comportamiento de los datos, como se tratará de explicar a continuación.

Iniciamos dando una breve explicación del método de Lagrange empleado en [4].

Dados

$$(x_i, f(x_i)) \text{ para cada } i = 0, 1, 2, \dots, n, \quad 2.1$$

si se define

$$f(x_i) = f_i ; \quad 2.2$$

la interpolación de Lagrange se obtiene en base a la siguiente expresión

$$P(x) = \sum_{i=0}^n f_i L_i(x) , \quad 2.3$$

donde

$$L_i(x) = \frac{(x-x_0) \cdots (x-x_{i-1})(x-x_{i+1}) \cdots (x-x_n)}{(x_i-x_0) \cdots (x_i-x_{i-1})(x_i-x_{i+1}) \cdots (x_i-x_n)} . \quad 2.4$$

¹ En el Capítulo anterior se mencionó que existen 42 parámetros en total en un Banco.

Con la formulación anterior veamos un ejemplo que demuestre la aseveración inicial, esto es, que los resultados obtenidos con el método de Lagrange en una variable dependen en gran medida del comportamiento de los datos.

Sean $(0,1)$, $(1,1)$, $(2,1)$ y $(2.5,2)$ puntos en el plano, si se aplica el método de Lagrange a dichos puntos se obtiene lo siguiente:

$$P(x) = 1 \cdot L_1(x) + 1 \cdot L_2(x) + 1 \cdot L_3(x) + 2 \cdot L_4(x), \quad 2.5$$

donde

$$\begin{aligned} L_1(x) &= \frac{(x-1)(x-2)(x-2.5)}{(0-1)(0-2)(0-2.5)}, \\ L_2(x) &= \frac{(x-0)(x-2)(x-2.5)}{(1-0)(1-2)(1-2.5)}, \\ L_3(x) &= \frac{(x-0)(x-1)(x-2.5)}{(2-0)(2-1)(2-2.5)}, \\ L_4(x) &= \frac{(x-0)(x-1)(x-2)}{(2.5-0)(2.5-1)(2.5-2)}, \end{aligned} \quad 2.6$$

simplificando se obtiene el siguiente polinomio de Lagrange

$$P_L(x) = \frac{8}{15}x^3 - \frac{24}{15}x^2 + \frac{16}{15}x + 1. \quad 2.7$$

En la Figura 2.1 se muestra la gráfica obtenida con el polinomio anterior, en ella también se muestran los datos proporcionados para la obtención del polinomio interpolante.

De la Figura 2.1 se puede observar que el comportamiento de los datos influye en un alto porcentaje en el polinomio final, esto es, se puede dar el caso que aún cuando los datos originales provengan de una función sin cambios notables, el polinomio de Lagrange genere muchas oscilaciones.

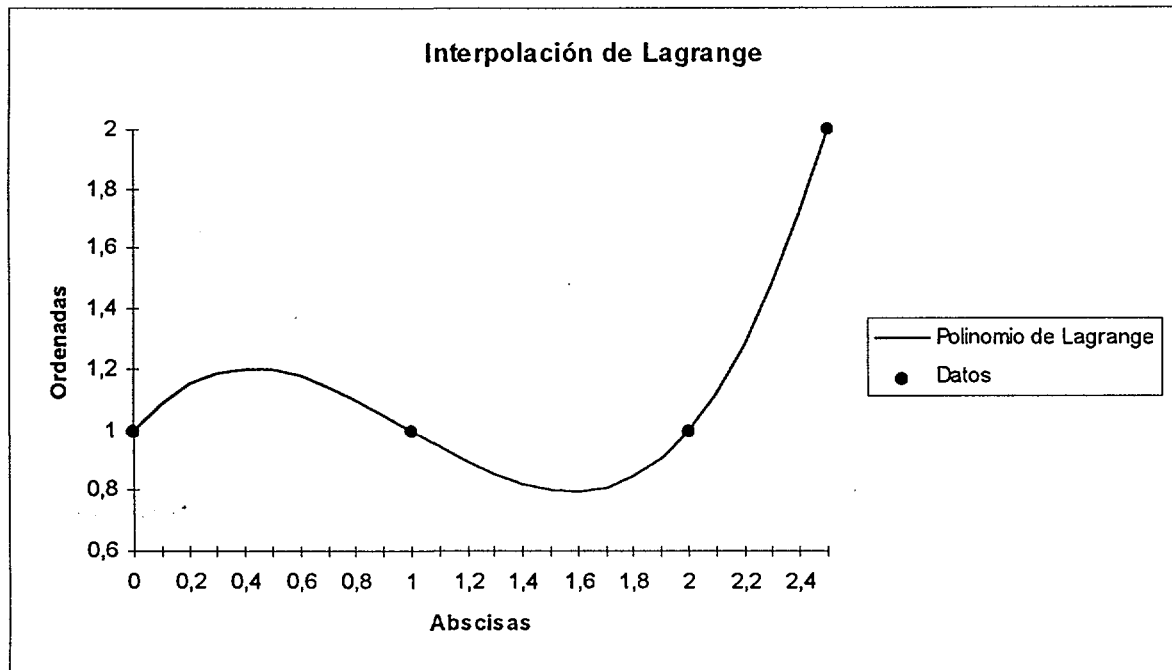


Figura 2.1. Comportamiento con el polinomio de Lagrange.

A continuación se muestra una situación que sin mayor análisis resulta obvia, y que sin embargo, dadas las implicaciones que esta tiene en el caso particular de los Bancos, es necesario puntualizarla.

Sean A(1,3), B(3,1) y C(5,3) puntos en el plano, se pueden obtener dos polinomios de Lagrange tomando en cuenta los puntos siguientes

- a) A, B y C
- b) A y C

Los polinomios que se obtienen en ambos casos son

$$P_1(x) = \frac{3}{8}x^2 - \frac{5}{2}x + \frac{41}{8}, \tag{2.8}$$

y

$$P_2(x) = -\frac{1}{4}x + \frac{13}{4}, \tag{2.9}$$

respectivamente; si los puntos A, B y C fueran datos reales, es obvio que los mejores resultados se obtiene empleando el polinomio 2.8, pues claramente 2.9 tiene un error considerable con respecto al punto B; la Figura 2.2 muestra la gráfica de ambos polinomios con los datos proporcionados. Esta situación que como ya se puntualizó resulta obvia, es un aspecto importante en el problema que se esta analizando debido a las observaciones que se harán en la siguiente sección.

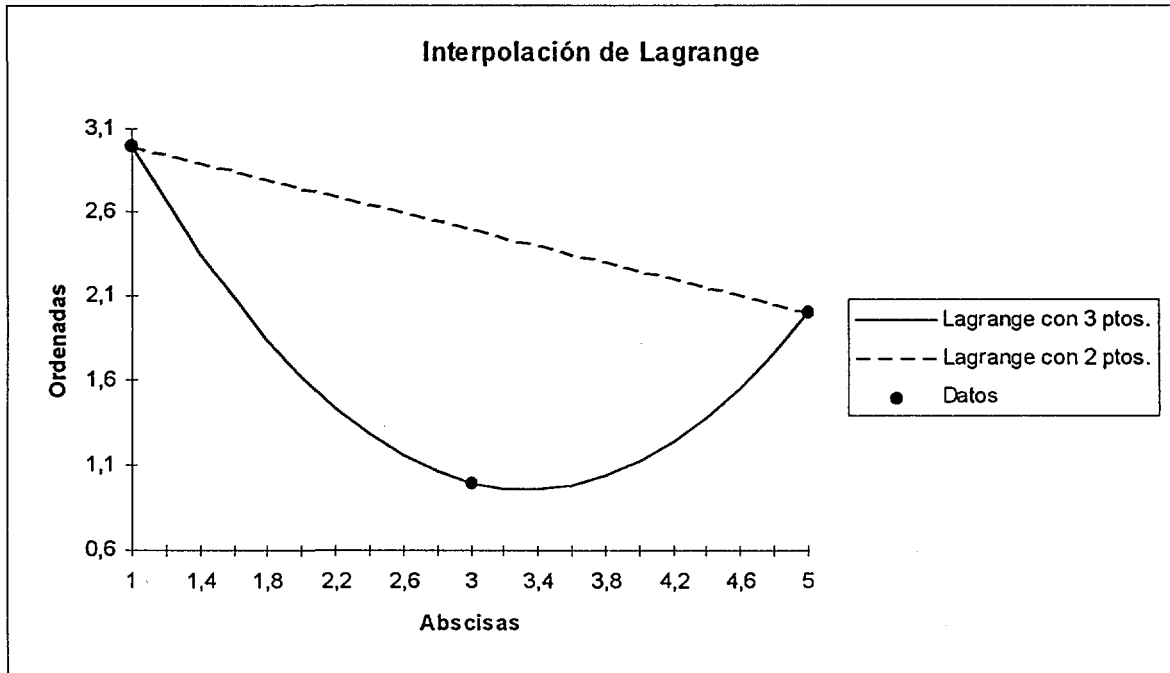


Figura 2.2. Interpolación de Lagrange tomando diferentes puntos.

2.2.- Observaciones

Con los ejemplos mostrados anteriormente se pueden hacer las siguientes observaciones, tomando en cuenta los Bancos de Datos Nucleares. Ya se ha mencionado que al obtener un nuevo Banco por medio de la interpolación, el problema se reduce a obtener tantos polinomios interpolantes como parámetros tenga el Banco, si se observa el primer ejemplo de la sección anterior, se puede afirmar que los resultados obtenidos dependerán en gran medida del comportamiento de cada uno de los parámetros que se van a interpolar, esto es, si alguno de los parámetros tiene un comportamiento como el mostrado en la Figura 2.1, los resultados no serán del todo adecuados. Por otra parte, el segundo ejemplo se mostró por lo siguiente, en el trabajo del IIE se afirma que basta con tomar dos Bancos de Datos Nucleares, a lo mas tres, para obtener buenos

resultados, sin embargo el ejemplo demuestra lo contrario, pues si se utilizan únicamente dos Bancos se está afirmando que el comportamiento de los parámetros es lineal, lo cual no es del todo cierto. Los resultados obtenidos pueden ser buenos al emplear los polinomios de Lagrange siempre y cuando se cumplan las dos condiciones siguientes:

- a) Que el comportamiento de los parámetros sea "*suave*"²
- b) Que si se desea interpolar empleando dos o tres Bancos, los valores empleados estén lo suficientemente cercanos³ para evitar errores.

Si se cumple lo anterior se puede afirmar que los resultados obtenidos serán bastante buenos, de hecho, se tiene la ventaja que utilizando Lagrange en esta situación, no requiere de mucho trabajo y es fácil de instrumentar.

Finalmente, es necesario tomar en cuenta las consideraciones hechas anteriormente, con el fin de saber en qué casos resulta apropiado utilizar dicha metodología y obtener buenos resultados. De hecho, se puede afirmar que son muchos los casos en los cuales se cumplen las condiciones antes mencionadas, por lo cual en esos casos resulta muy útil la metodología expuesta. En el último capítulo se abundará un poco más sobre la diferencia de los resultados obtenidos con el método de Lagrange en una variable y la propuesta hecha en el presente trabajo, al aplicarlos a diferentes ejemplos.

² El ejemplo muestra que se puede entender como un comportamiento "*no suave*".

³ Se refiere a los valores de los parámetros de los Bancos, pues son los que se utilizan para la interpolación.

Capítulo 3

Comportamiento de los Parámetros

En este capítulo se da una explicación de las ideas básicas que surgieron al resolver el problema planteado originalmente, utilizando MATLAB, en éste se realizaron las primeras pruebas acerca del comportamiento de los parámetros más importantes de un Banco de Datos Nucleares; de las pruebas realizadas en MATLAB se llegó a la conclusión de emplear la interpolación polinomial bicúbica, la cual se abordará en un capítulo posterior. En este caso, el análisis consistió en emplear un spline bidimensional cúbico con los parámetros mencionados, posteriormente se obtuvieron las gráficas de dichos parámetros y al final se hicieron observaciones acerca de su comportamiento.

Conviene señalar que la idea de verificar, y al mismo tiempo obtener conclusiones acerca del comportamiento de los parámetros mas importantes de un Banco, se debió en primer lugar, a la problemática observada al emplear los polinomios de Lagrange; en segundo término, se consideró importante observar la parte

fundamental de un Banco, esto es, los parámetros que determinan en mayor grado su confiabilidad.

Por otra parte, ya se había vislumbrado la idea de emplear dos variables al llevar a cabo la interpolación, pues con ello se aprovecharía la mayor información que se tenía disponible; por esta razón, al aplicar el spline se tomaron en cuenta dichas variables, esto es, el uranio y la gadolinia.

3.1.- Parámetros y Variables

Como ya se ha mencionado, la generación de Bancos por medio de la interpolación surge del problema de encontrar un nuevo Banco a partir de otros que ya se tienen, es decir, interpolar cada uno de los parámetros que lo forman, en base a los diferentes quemados que se manejan. Podemos acotar aún más el problema inicial, si se toman en cuenta los siguientes factores: primero, observar que una parte de la información no cambia si las celdas de las que provienen los Bancos son del mismo tipo¹; por otro lado, la mayor parte de los parámetros a interpolar dependen de dos variables importantes, las cuales son los porcentajes de uranio (^{235}U) y gadolinia (Gd_2O_3).

Como se ha visto hasta el momento, la idea fue reducir el problema hasta donde sea posible, sin que esto conlleve a la obtención de resultados erróneos. Siguiendo con esta idea, ya se ha mencionado que de los parámetros que componen al Banco, 12 son los más importantes, pues son los que proporcionan las secciones eficaces macroscópicas de la celda y la k_{inf} .

El paso siguiente consistió en tomar un conjunto de Bancos del mismo tipo y en los cuales se tuvieran diferentes valores para el ^{235}U y la Gd_2O_3 , con esta información se procedió a obtener las gráficas de cada uno de los parámetros anteriores, con la idea de poder observar el comportamiento que tenían.

Es necesario hacer algunas observaciones acerca de las gráficas obtenidas: los asteriscos representan los valores de los parámetros, la malla que se presenta se obtiene discretizando en cada uno de los ejes que se toman en cuenta; el eje X corresponde a la gadolinia y el eje Y corresponde al uranio; en el eje Z se grafica el

¹ Se dice que dos celdas son del mismo tipo si su geometría y distribución de materiales son las mismas.

correspondiente valor del parámetro. En la parte superior de cada gráfica se indica el quemado correspondiente.

Por cuestiones de espacio no es posible presentar todas las gráficas obtenidas en el proceso antes mencionado, por tal motivo solamente se presentaran un número reducido de ellas, con la observación de que el comportamiento presentado es similar en todos los casos. Es necesario remarcar que se obtuvo un buen número de gráficas antes de dar una conclusión al respecto.

A continuación se presentan 12 gráficas (Figuras 3.1 a 3.12) donde se muestran los siguientes parámetros: sección eficaz macroscópica, número de neutrones producidos por fisión en el grupo 2, sección eficaz macroscópica de absorción y el coeficiente de difusión; cada parámetro con quemados de 0, 10, 15 y en algunos casos 20 MWd/t respectivamente. Los datos provienen de Bancos del tipo G-9, con una geometría 8x8 y con canal central de agua².

En la Tabla 3.1 se muestran los porcentajes de gadolinia y uranio empleados para obtener las gráficas antes mencionadas.

Porcentajes de uranio y gadolinia empleados				
Gd ₂ O ₃	7	14	21	28
²³⁵ U	3.6091	3.7091	3.8091	3.9091

Tabla 3.1. Porcentajes de ²³⁵U y Gd₂O₃.

Es necesario aclarar que en el caso del uranio el valor que se toma en cuenta es el porcentaje en peso del contenido de ²³⁵U, mientras que para la gadolinia se toman siete barras con porcentajes de 1, 2, 3 y 4 % respectivamente, resultando con ello los valores mostrados en la tabla 3.1, al realizar la multiplicación por siete³.

² Los ensambles de combustible del tipo señalado son fabricados por General Electric Co.

³ Al hablar del porcentaje de gadolinia en un ensamble de combustible se toma en cuenta el porcentaje de una barra multiplicado por el número de éstas que contienen dicho elemento.

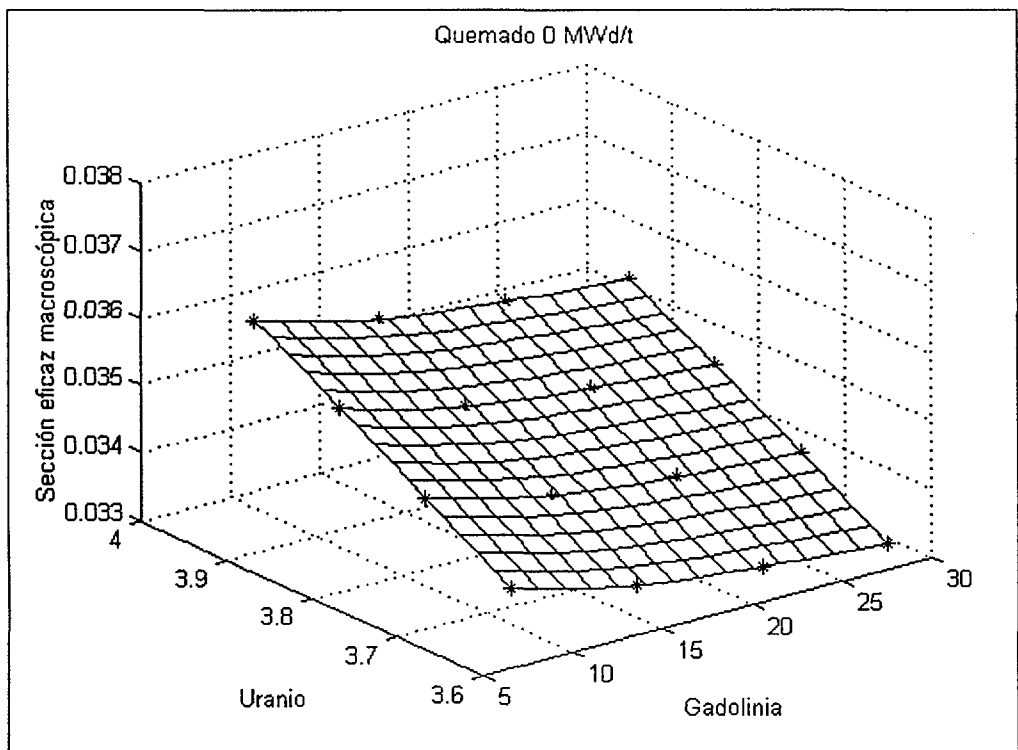


Figura 3.1. Comportamiento de Σ_f con quemado 0.

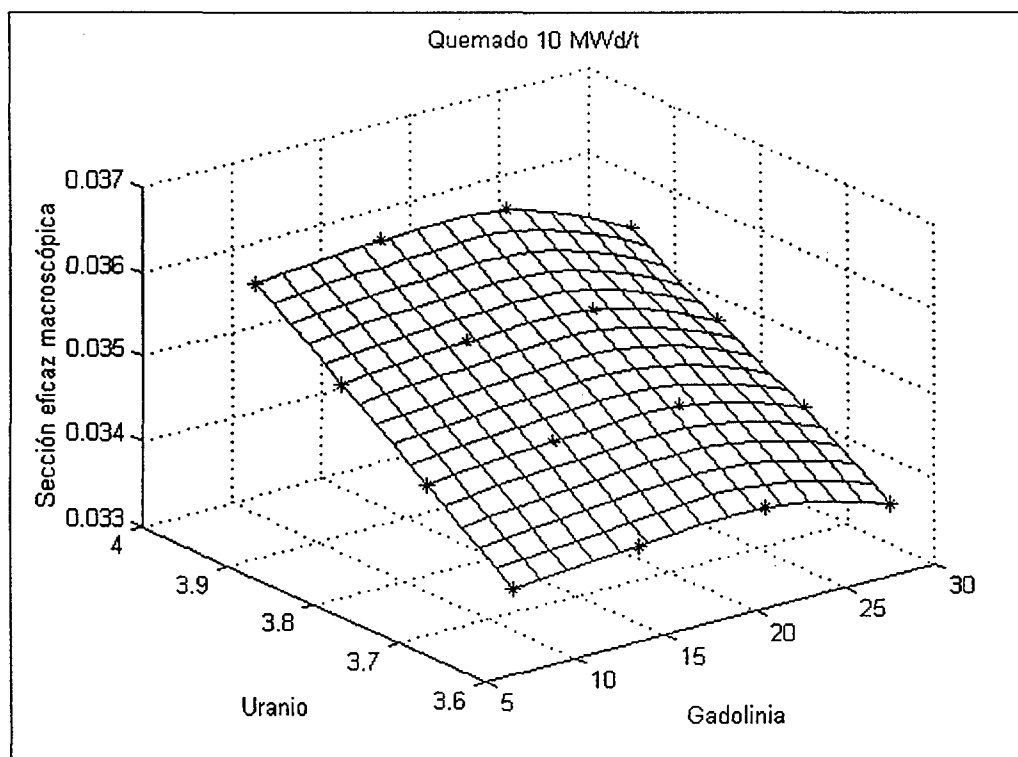


Figura 3.2. Comportamiento de Σ_f con quemado 10.

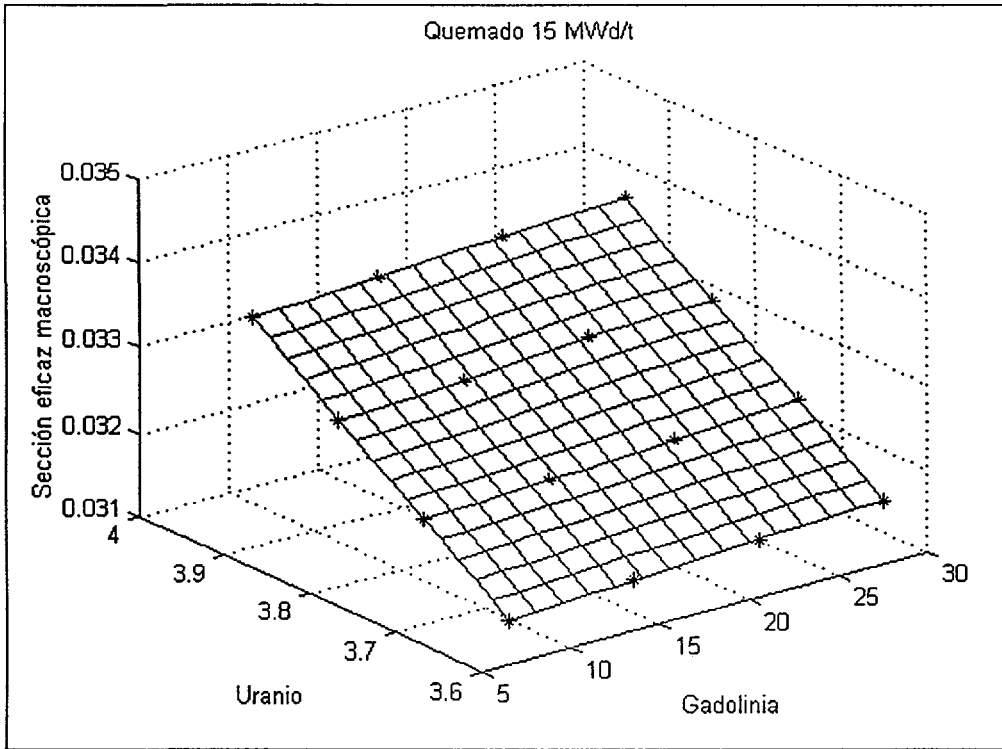


Figura 3.3. Comportamiento de Σ_{f2} con quemado 15.

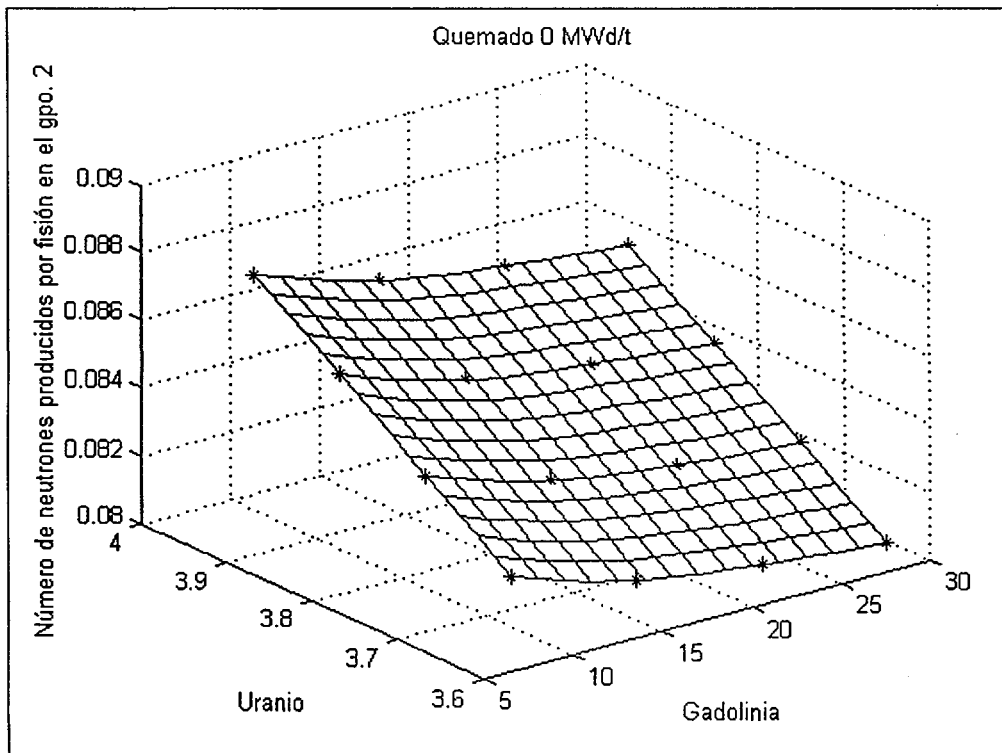


Figura 3.4. Comportamiento de $\nu\Sigma_{f2}$ con quemado 0.

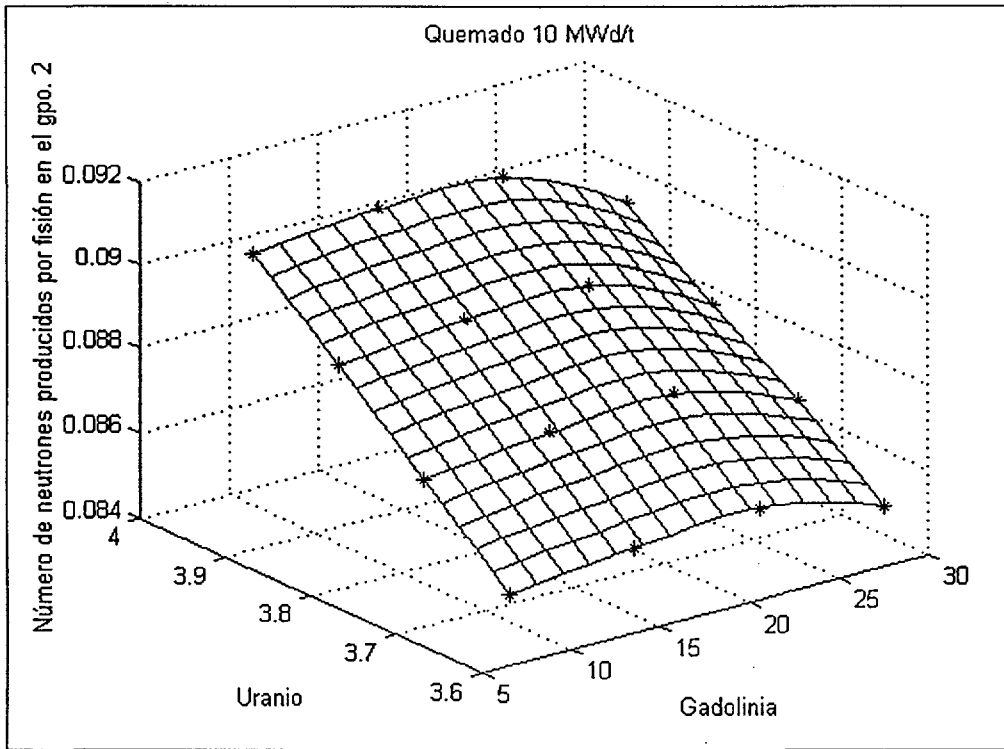


Figura 3.5. Comportamiento de $\nu\Sigma_{f2}$ con quemado 10.

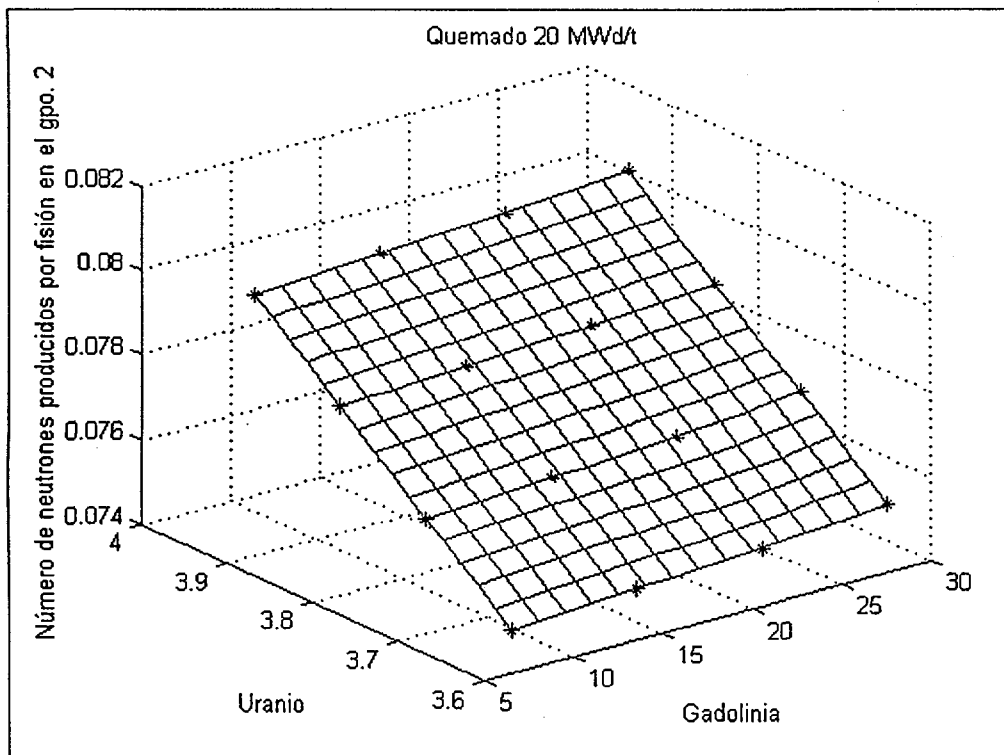


Figura 3.6. Comportamiento de $\nu\Sigma_{f2}$ con quemado 20.

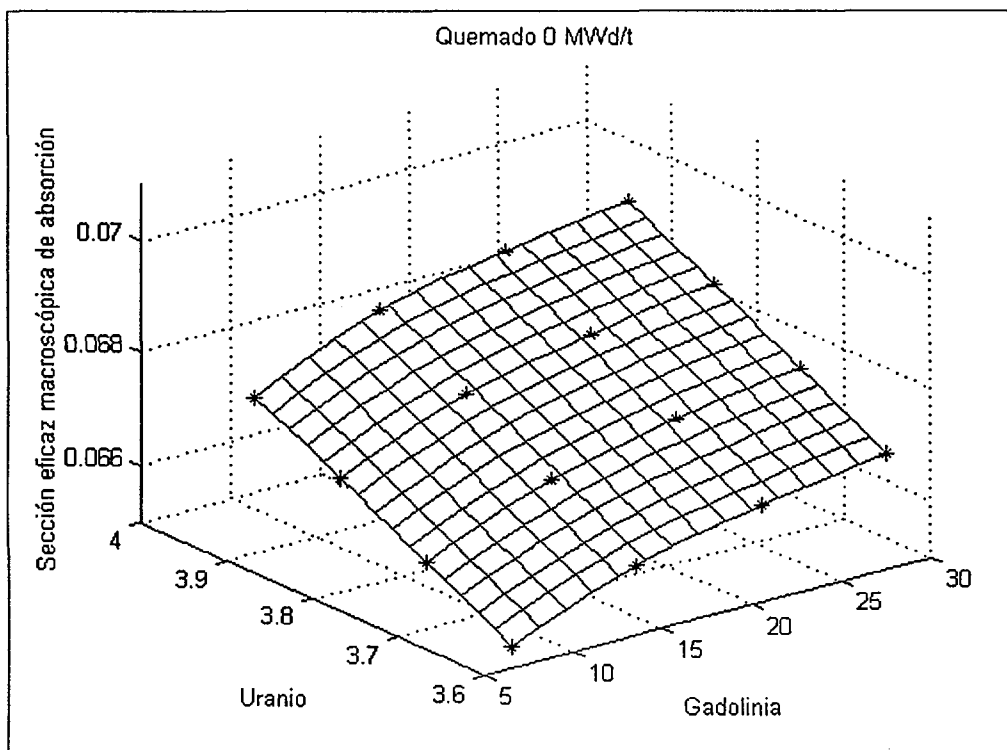


Figura 3.7. Comportamiento de Σ_{a2} con quemado 0.

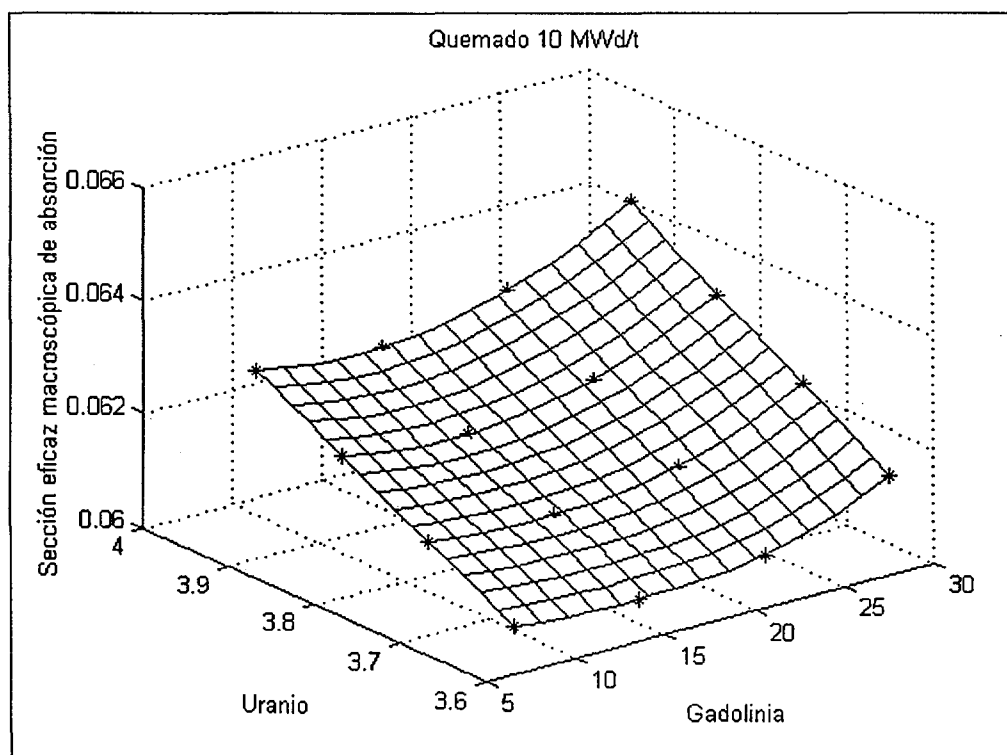


Figura 3.8. Comportamiento de Σ_{a2} con quemado 10.

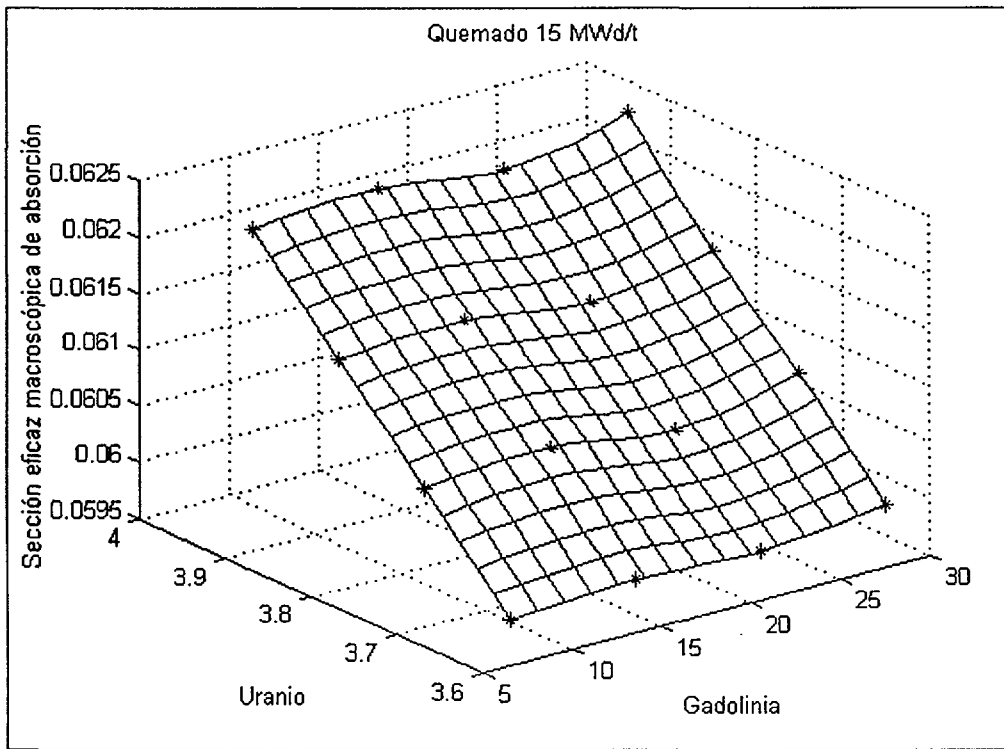


Figura 3.9. Comportamiento de Σ_{a2} con quemado 15.

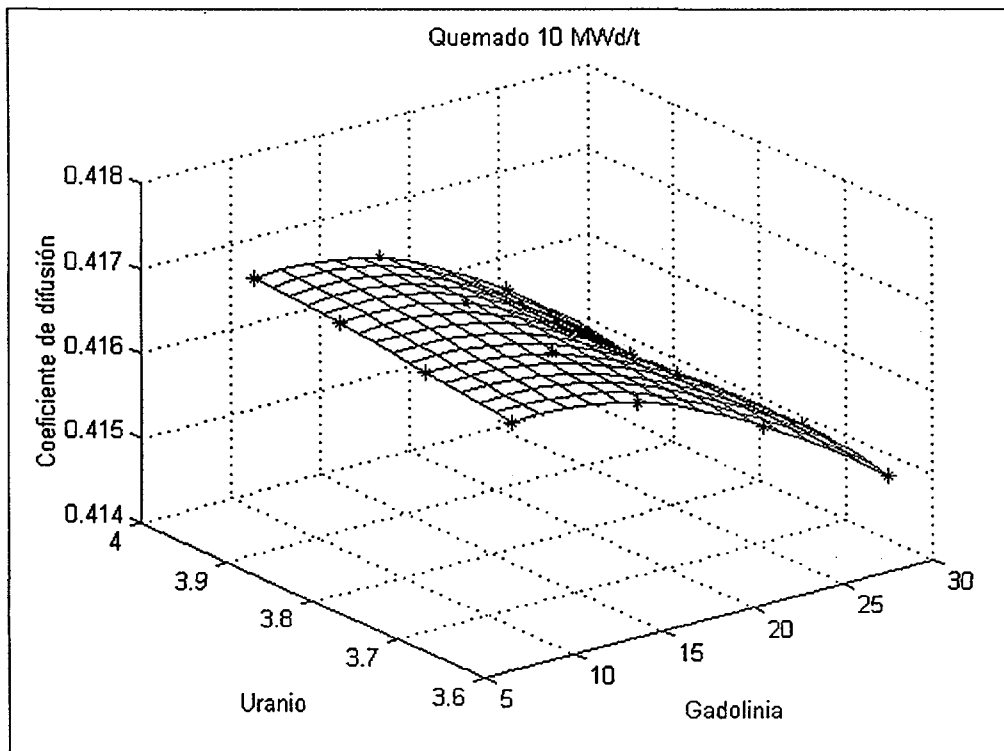


Figura 3.10. Comportamiento de D_2 con quemado 10.

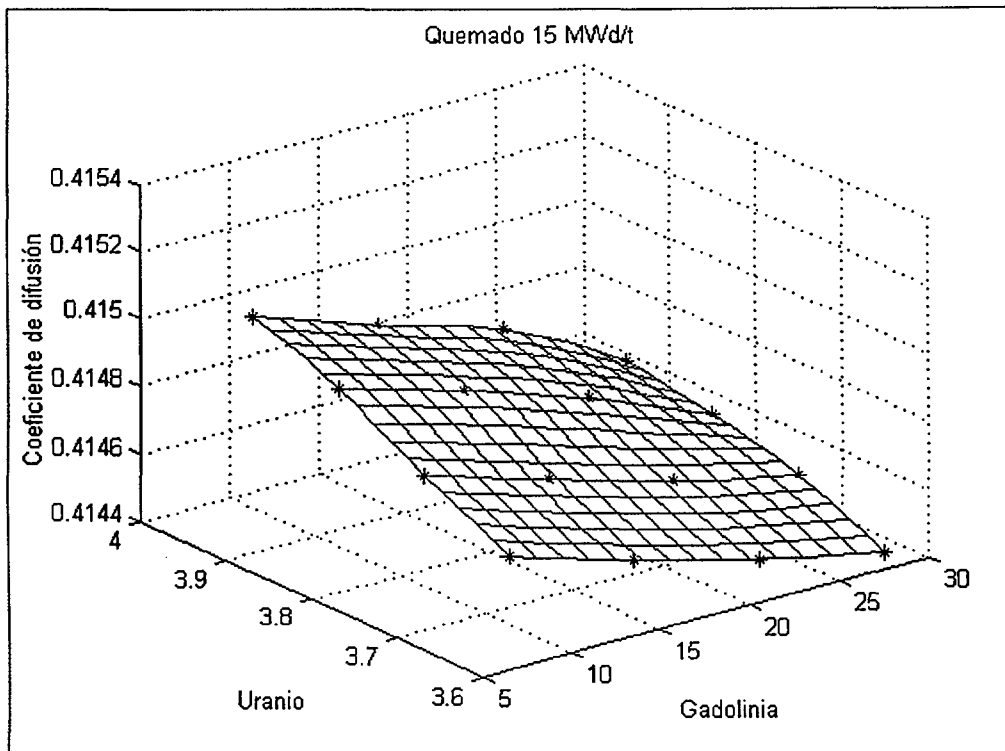


Figura 3.11. Comportamiento de D_2 con quemado 15.

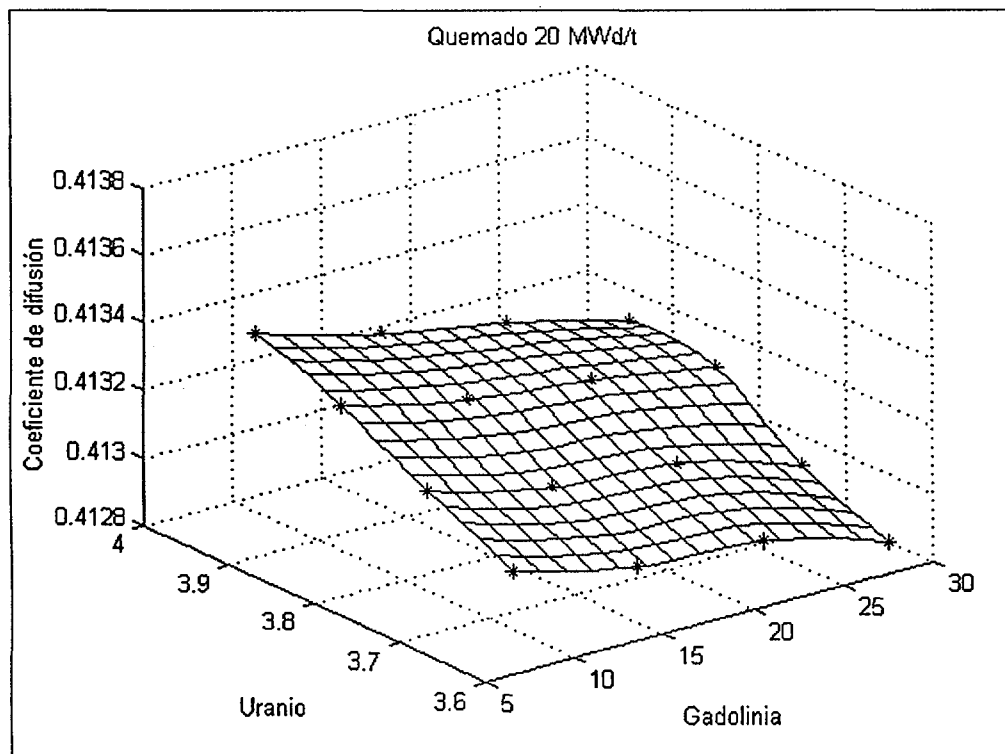


Figura 3.12. Comportamiento de D_2 con quemado 20.

3.2.- Comportamiento de las Gráficas

Es necesario remarcar que las conclusiones que a continuación se presentan, se basan tomando en cuenta el total de gráficas obtenidas, de una buena cantidad de Bancos de los cuales se tenía la información respectiva, no únicamente de las gráficas mostradas anteriormente.

De las gráficas obtenidas se pueden hacer las siguientes observaciones:

- a) No existen comportamientos bruscos en cada una de las gráficas analizadas.
- b) En algunos casos el comportamiento es de tipo cuadrático o lineal.
- c) En la mayoría de los casos, el comportamiento en la variable x (uranio) es creciente.

De las observaciones anteriores se pueden obtener conclusiones importantes al respecto: en primer lugar no es necesario proponer una función complicada para el comportamiento de los parámetros, debido a que el comportamiento de los mismos es "*suave*" en ambas direcciones; existen pequeñas oscilaciones en algunos parámetros sin que estas lleguen a picos pronunciados, sin embargo, conviene aprovechar la mayor información posible con la cual se cuenta.

En un primer acercamiento a la solución del problema, se pensó que una función bilineal podría ser suficiente para aproximarse de manera adecuada al comportamiento observado, sin embargo, debido a que en algunos casos se presentan pequeñas oscilaciones, se puede afirmar que una propuesta viable para llevar a cabo la interpolación es un polinomio bicúbico en ambas variables, el cual se analizará en el siguiente capítulo con mas detalle.

Para terminar, conviene señalar que los valores del uranio fluctúan entre 2 y 4 %, razón por la cual no se puede pensar en cambios notables en dicho porcentaje, que pudieran repercutir de manera importante en el comportamiento de un Banco. Análogamente, los valores para la gadolinia se mantienen entre 5 y 35 %⁽⁴⁾, por lo cual tampoco se esperan cambios notables en este sentido.

⁴ Tomando en cuenta que dicho valor se obtiene multiplicando el número de barras que existen con dicho componente.

Capítulo 4

Interpolación

Polinomial Bicúbica

Hasta el momento se han construido las bases necesarias para resolver de manera adecuada el problema planteado originalmente, esto es, contar con una herramienta confiable para la obtención de un nuevo Banco de Datos Nucleares a partir de otros ya existentes, lo cual permita ahorrar recursos, tanto humanos como de cómputo.

En los capítulos anteriores se analizó la estructura de los Bancos, además de su generación y el comportamiento de los parámetros más importantes involucrados, estos últimos a diferentes quemados. Con todo esto, se hizo una propuesta viable que resolvió de manera satisfactoria la problemática planteada.

En este capítulo se proporciona el desarrollo matemático empleado en la interpolación polinomial bicúbica. Para esto, de manera muy particular se dan las ideas básicas del método de elemento finito, aplicado éste a la interpolación en dos variables. Posteriormente se aplican dichas ideas en la solución del problema en cuestión. Finalmente, se hace referencia al número de condición de la matriz de

interpolación, debido a que este surge en la solución de la gran cantidad de sistemas lineales que se necesitan para hallar los polinomios interpolantes¹.

De lo anterior, es importante mencionar lo siguiente; en la introducción se mencionó que una vez concluida la primera versión del código *INTPOLBI*, se continuó trabajando con la idea de mejorar éste. En ese sentido, se modificaron algunos procesos por otros más eficientes, particularmente se modificó el método numérico para encontrar la solución de los sistemas de ecuaciones lineales mencionados previamente, obteniendo con ello mejores resultados. En este capítulo se presenta el desarrollo empleado en ambas versiones mencionando las razones por las cuales los resultados son mejores. Los resultados obtenidos, así como el análisis hecho a los mismos se deja para el capítulo siguiente.

4.1.- Interpolación con Elemento Finito

En analogía al caso unidimensional, cuando se trabaja en \mathcal{R}^2 , la idea principal consiste en lo siguiente: dada una función f definida en un intervalo $[a,b] \times [c,d]$, se desea construir una aproximación de elemento finito (interpolante) g_h de f , que coincida con f en ciertos puntos; para esto se divide el intervalo dado en una colección de elementos finitos o malla. En otras palabras, el método consiste en aproximar una función continua por medio de un modelo discreto, compuesto por un conjunto finito de funciones continuas, definidas sobre un número finito de subdominios. El método se puede resumir en los cinco pasos siguientes:

1. Identificar un número finito de puntos en el dominio. A dichos puntos se les conoce como "nodos" o puntos nodales.
2. Denotar el valor de la función continua en cada nodo como una variable, la cual va a ser determinada.
3. Dividir el dominio en un número finito de subdominios llamados elementos. Estos elementos se conectan a los nodos comunes y en conjunto se aproximan a la forma del dominio original.
4. Aproximar la función continua sobre cada elemento por medio de un polinomio que se define por medio de los valores en los nodos de la función continua. Se define un polinomio para cada elemento, tomando en cuenta que los polinomios son seleccionados de tal manera que la continuidad se mantenga a lo largo de las fronteras de los elementos.

¹ No se debe olvidar que se encuentran tantos polinomios como parámetros existen en el Banco.

5. Validar las características de la solución obtenida y, de ser posible, calcular una estimación del error.

Es necesario formalizar lo anterior. Sea Ω un dominio en \mathcal{R}^2 , en general se busca una malla $\Omega_i, i=1, \dots, M$ de elementos finitos ya sea triangulares o cuadriláteros que cubran a Ω , si $\partial\Omega$ es curva, la colección Ω_i no coincidirá perfectamente con el dominio de Ω dado, esta situación se presenta en la Figura 4.1. Obviamente la malla puede ser refinada en la medida de los recursos disponibles y las necesidades. Por facilidad en la escritura, a la unión de los $\Omega_i, i=1, \dots, M$ la denotaremos por Ω_h .

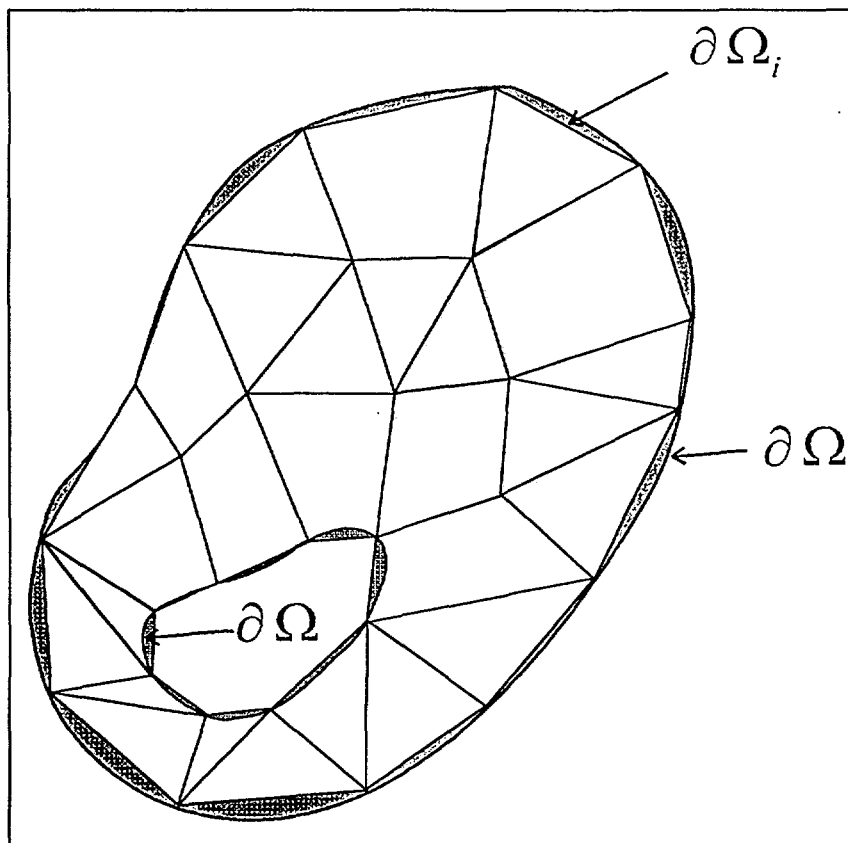


Figura 4.1. Una discretización de elemento finito.

Conviene aclarar que una razón por la cual se eligen formas simples, como triángulos o cuadriláteros, es la existencia de una correspondencia natural entre el número de nodos en un elemento y el número de términos usados en la aproximación polinomial. Por ejemplo, al usar triángulos se tiene que un polinomio completo de grado k en x y y tiene exactamente $\frac{1}{2}(k+1)(k+2)$ términos. Por lo tanto,

un polinomio de grado k se puede determinar especificando su valor en $\frac{1}{2}(k+1)(k+2)$ puntos en el plano, que no son otra cosa que el número de nodos elegidos para el elemento. En la Figura 4.2 se muestra dicha correspondencia.

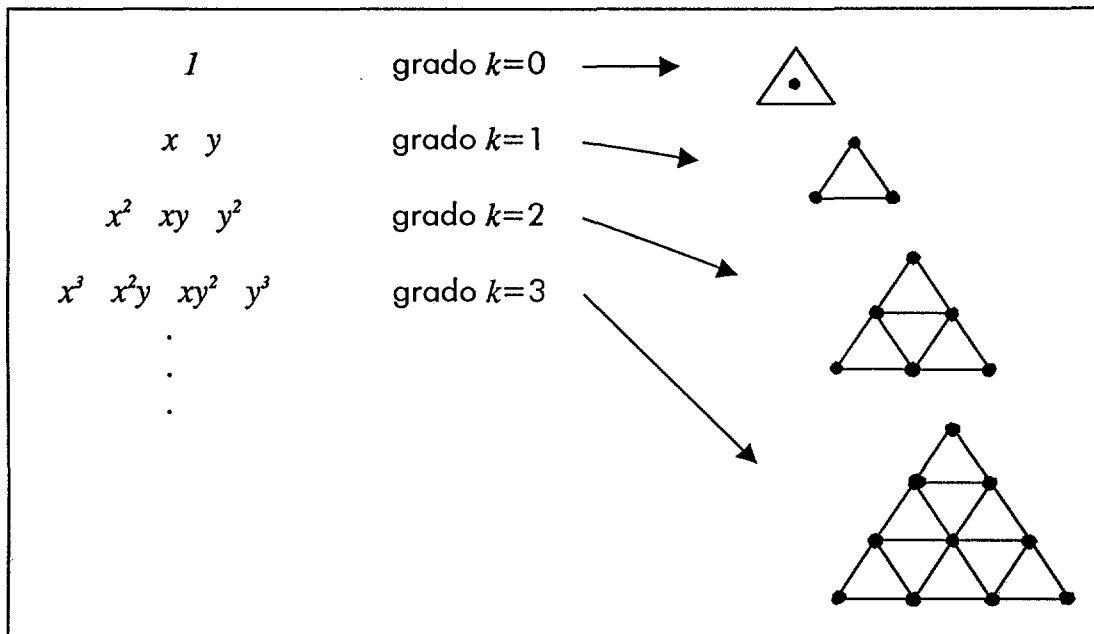


Figura 4.2. Relación entre el número de términos del interpolante, los elementos y sus nodos.

La idea ahora es proporcionar los detalles de cómo se construyen dichas representaciones de elemento finito de dos dimensiones, sin olvidar que el concepto de elemento finito es un medio para interpolar una función dada $f = f(x,y)$ definida sobre Ω .

Veamos, para cada elemento de la malla es necesario construir un interpolante g_h de f de la forma:

$$g_h(x,y) = \sum_{j=1}^N g_j \phi_j(x,y), \tag{4.1}$$

donde N es el número de nodos elegidos² para el elemento y $(x,y) \in \Omega_i$, además se tiene que

² El número de nodos depende del elemento seleccionado.

$$\phi_1(x,y), \phi_2(x,y), \dots, \phi_N(x,y), \quad 4.2$$

son funciones definidas sobre Ω_i que satisfacen

$$\phi_i(x_j, y_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j, \end{cases} \quad 4.3$$

donde (x_j, y_j) son las coordenadas de los puntos en la malla de elemento finito. Si 4.3 se cumple, entonces se tiene que

$$g_h(x_j, y_j) = g_j, \quad j = 1, 2, \dots, N, \quad 4.4$$

de tal forma que si se pide

$$g_j = f(x_j, y_j) \quad 4.5$$

g_h coincidirá (aún más, interpolará) con la función dada f en los puntos de la malla.

En particular, para una subregión triangular en dos dimensiones se pueden tener diferentes funciones dependiendo del número de nodos que se quiera trabajar, por ejemplo, una función lineal tiene la forma:

$$g_h(x, y) = a_1 + a_2x + a_3y, \quad 4.6$$

de donde, con tres valores independientes de g_h , es decir tres nodos, se pueden determinar los valores a_1 , a_2 y a_3 .

Para determinar los valores de a_1 , a_2 y a_3 , de 4.4 se tiene que

$$\begin{aligned} g_1 &= g_h(x_1, y_1) = a_1 + a_2x_1 + a_3y_1, \\ g_2 &= g_h(x_2, y_2) = a_1 + a_2x_2 + a_3y_2, \\ g_3 &= g_h(x_3, y_3) = a_1 + a_2x_3 + a_3y_3, \end{aligned} \quad 4.7$$

donde (x_i, y_i) , $i=1, \dots, 3$, son las coordenadas de los tres vértices del triángulo, es decir, los nodos del elemento. Resolviendo el sistema para a_1 , a_2 y a_3 , se tiene que

$$\begin{aligned} a_1 &= [g_1(x_2y_3 - x_3y_2) + g_2(x_3y_1 - x_1y_3) + g_3(x_1y_2 - x_2y_1)]/(2A), \\ a_2 &= [g_1(y_2 - y_3) + g_2(y_3 - y_1) + g_3(y_1 - y_2)]/(2A), \\ a_3 &= [g_1(x_3 - x_2) + g_2(x_1 - x_3) + g_3(x_2 - x_1)]/(2A), \end{aligned} \quad 4.8$$

donde

$$A = x_2y_3 + x_1y_2 + x_3y_1 - y_1x_2 - y_2x_3 - y_3x_1. \quad 4.9$$

Sustituyendo los valores de a_1 , a_2 y a_3 en 4.6, se tiene que

$$g_h(x, y) = g_1\phi_1(x, y) + g_2\phi_2(x, y) + g_3\phi_3(x, y), \quad 4.10$$

donde

$$\begin{aligned} \phi_1 &= [(x_2y_3 - x_3y_2) + (y_2 - y_3)x + (x_3 - x_2)y]/(2A), \\ \phi_2 &= [(x_3y_1 - x_1y_3) + (y_3 - y_1)x + (x_1 - x_3)y]/(2A), \\ \phi_3 &= [(x_1y_2 - x_2y_1) + (y_1 - y_2)x + (x_2 - x_1)y]/(2A). \end{aligned} \quad 4.11$$

Es necesario notar que

$$\phi_i(x_j, y_j) = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j, \end{cases} \quad i, j = 1, 2, 3. \quad 4.12$$

Por otro lado, es importante mencionar que se debe poner un énfasis especial en la discretización del dominio Ω , es decir, qué sucede con cada uno de los subdominios de Ω_i definidos para crear la malla de elementos finitos, particularmente en la frontera de dos elementos contiguos, en dicho caso se debe cumplir que, para cualesquiera dos polinomios que provengan de dos elementos contiguos, su frontera sea continua. En la Figura 4.3 se muestra un ejemplo de dos elementos triangulares que cumplen tal situación.

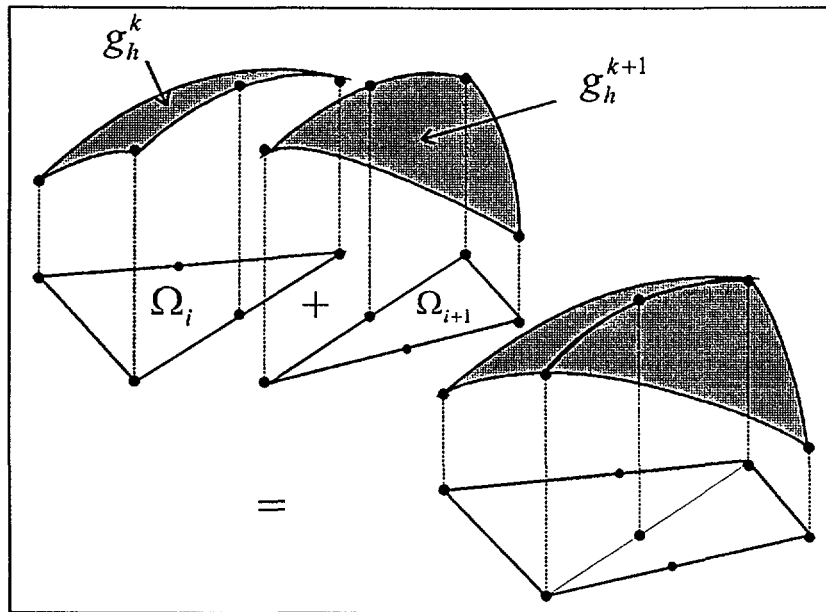


Figura 4.3. Frontera de dos elementos contiguos.

Para que se cumpla lo anterior es necesario que

$$\int_{\Omega_i} \left[\left(\frac{\partial \phi_i}{\partial x} \right)^2 + \left(\frac{\partial \phi_i}{\partial y} \right)^2 + \phi_i^2 \right] dx dy < \infty, i=1, \dots, N, \quad 4.13$$

la expresión anterior no es otra cosa que una norma definida en un espacio de Sobolev y nos indica que las funciones ϕ_i propuestas deben cumplir que:

- a) Sean continuas.
- b) Sean acotadas.
- c) Las primeras derivadas sean continuas en el dominio.
- d) Sean lo suficientemente "suaves".

En el caso analizado, las funciones elegidas para 4.3 son polinomios, por lo tanto, todas las situaciones anteriores se cumplen cabalmente.

Para finalizar el desarrollo anterior veamos ahora lo siguiente; al aplicar el método de elemento finito surgen dos clases de errores, los usuales de truncamiento y redondeo asociados a la parte computacional, y los relacionados a la técnica en si misma, estos últimos se pueden a su vez dividir de la manera siguiente:

- a) Errores de discretización, provenientes de la diferencia geométrica entre las fronteras de la región y su aproximación de elemento finito.
- b) Errores de la función, provenientes de la diferencia entre la solución real y la obtenida con el polinomio propuesto.

Los errores de discretización se pueden reducir usando elementos más pequeños y eventualmente, tender a cero dicho tamaño.

En el segundo caso, veamos qué sucede. Se asume que se desea interpolar a la función original por una representación de elementos finitos g_h , la cual contiene polinomios de grado k . Si las derivadas parciales de la función original de orden $k+1$ son acotadas en Ω_i , el error de interpolación satisface que [6]

$$\|f - g_h\|_{\infty, \Omega_i} = \max_{(x,y) \in \Omega_i} |f(x,y) - g_h(x,y)| \leq Ch_i^{k+1}, \quad 4.14$$

donde C es una constante positiva y h_i es el "diámetro" de Ω_i ; esto es, la distancia más grande entre dos puntos de Ω_i . Es necesario hacer notar que esta estimación se mantiene únicamente si todos los términos en el polinomio de grado k aparecen en g_h . Además,

$$\left\| \frac{\partial f}{\partial x} - \frac{\partial g_h}{\partial x} \right\|_{\infty, \Omega_i} \leq C_1 h_i^k \quad \text{y} \quad \left\| \frac{\partial f}{\partial y} - \frac{\partial g_h}{\partial y} \right\|_{\infty, \Omega_i} \leq C_2 h_i^k. \quad 4.15$$

Por otro lado, como ya se mencionó anteriormente, la expresión siguiente es una norma en dos dimensiones, la cual denotaremos con el subíndice "s", esto es

$$\|f\|_s^2 = \int_{\Omega} \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 + f^2 \right] dx dy \quad 4.16$$

es una norma en dos dimensiones. Supongamos que $\Omega_i = \Omega$ y h es el diámetro máximo de todos los elementos en la malla de elemento finito generada. Se puede mostrar que para h suficientemente pequeña se cumple que [6]

$$\|f - g_h\|_s \leq C_3 h^k .$$

4.17

Es importante señalar que esta relación se mantiene si g_h contiene un polinomio completo de grado k . Por ejemplo, si se tienen elementos finitos de forma rectangular, para $k=3$, el polinomio es completo hasta grado 2, por esta razón los términos que aparecen de grado 3 contribuirán a la continuidad de la interpolación, sin embargo, no contribuirán en la convergencia del error de interpolación [6].

4.2.- Aplicación de la Técnica a un Banco

La idea en este momento es aplicar la técnica de elemento finito al caso particular de los Bancos de Datos Nucleares. Para avanzar de manera ordenada, primero se recordarán algunas ideas importantes desde el punto de vista nuclear por dos razones esenciales, primero para no perder de vista los detalles finos de éste, y en segundo término, para analizar la manera de relacionar la base teórica, desde el punto de vista matemático y el problema a resolver.

Con el fin de no perder de vista la base teórica, se debe recordar que once de los parámetros más importantes en un Banco son los siguientes:

- 1) Coeficiente de difusión del Grupo 1 (D_1)
- 2) Coeficiente de difusión del Grupo 2 (D_2)
- 3) Sección eficaz de absorción del grupo 1 (Σ_{A1})
- 4) Sección eficaz de absorción del grupo 2 (Σ_{A2})
- 5) Sección eficaz de remoción del grupo 1 al grupo 2 ($\Sigma_{R1,2}$)
- 6) Número de neutrones producidos por fisión en el grupo 1 ($\nu\Sigma_{f1}$)
- 7) Número de neutrones producidos por fisión en el grupo 2 ($\nu\Sigma_{f2}$)
- 8) Sección eficaz de fisión del grupo 1 (Σ_{f1})
- 9) Sección eficaz de fisión del grupo 2 (Σ_{f2})
- 10) Espectro de fisión del grupo 1 (X_1)
- 11) Espectro de fisión del grupo 2 (X_2)

Del análisis realizado, tanto de la técnica de elemento finito, como del comportamiento de los once parámetros anteriores, veamos cómo relacionar ambas cosas.

Ya se ha mencionado que la malla de elemento finito que se forma está compuesta por "*elementos*" simples, tales como triángulos, cuadriláteros o alguna combinación

de ellos, los que a su vez tienen nodos en los cuales la función propuesta coincide con la función original. Para obtener los polinomios interpolantes existen diferentes caminos dependiendo de la forma de los elementos.

En este trabajo, para el cual se eligieron rectángulos, los polinomios de interpolación se obtienen por medio de un producto tensorial. En la Tabla 4.1 se muestran dichos productos para distintos órdenes en los polinomios interpolantes.

	1	y	y^2	y^3	y^4	\dots
1	1	y	y^2	y^3	y^4	\dots
x	x	xy	xy^2	xy^3	xy^4	\dots
x^2	x^2	x^2y	x^2y^2	x^2y^3	x^2y^4	\dots
x^3	x^3	x^3y	x^3y^2	x^3y^3	x^3y^4	\dots
x^4	x^4	x^4y	x^4y^2	x^4y^3	x^4y^4	\dots
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

Tabla 4.1. Producto tensorial para obtener los polinomios en el caso de rectángulos.

La Tabla 4.1 incluye los términos necesarios dependiendo del número de nodos en el elemento para el caso de rectángulos, la correspondencia entre el número de nodos y los términos del polinomio se muestra en la Figura 4.4.

En particular, para obtener el polinomio de interpolación en el caso de los Bancos de Datos Nucleares, se eligió un elemento rectangular con 16 nodos. De la tabla 4.1 se tiene el siguiente desarrollo.

Con el producto tensorial

$$\begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix} \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \tag{4.18}$$

se obtiene la matriz

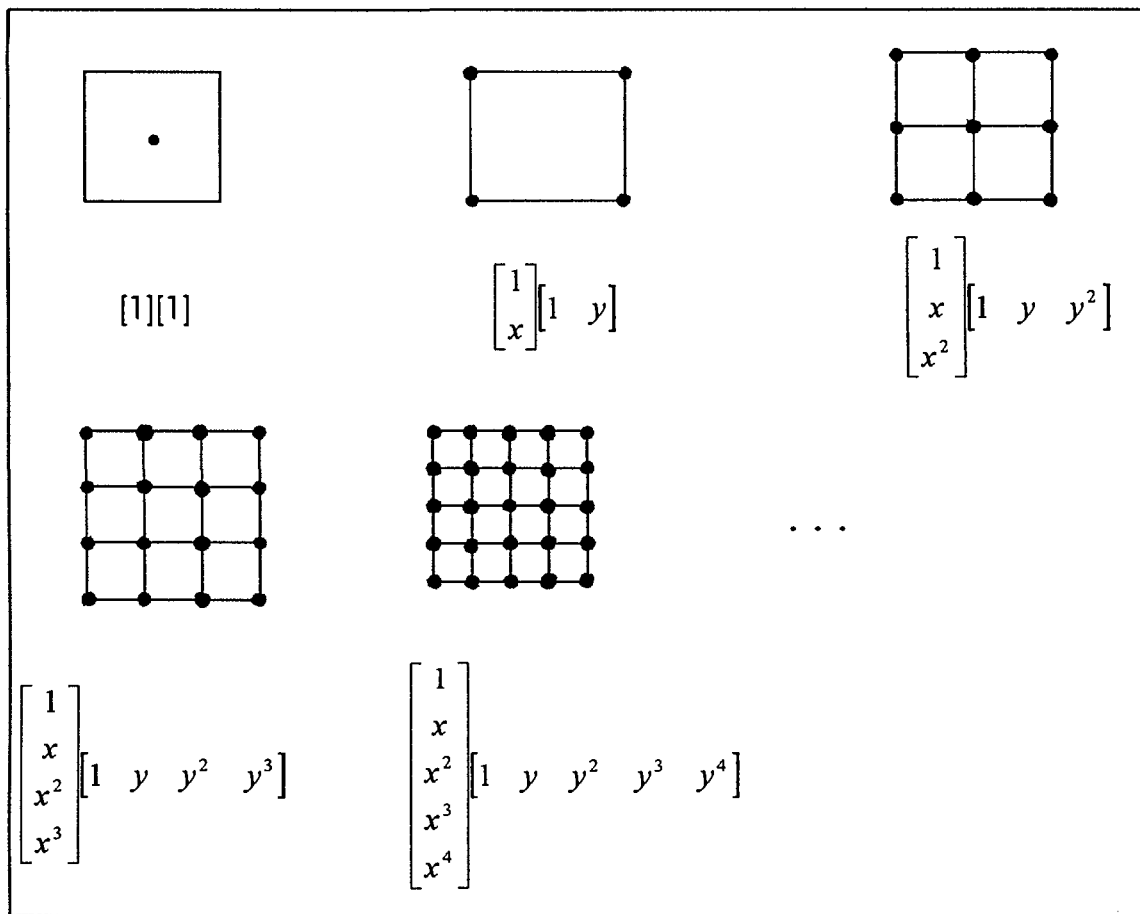


Figura 4.4. Producto tensorial para elementos rectangulares.

$$\begin{bmatrix} 1 & x & x^2 & x^3 \\ y & xy & x^2y & x^3y \\ y^2 & xy^2 & x^2y^2 & x^3y^2 \\ y^3 & xy^3 & x^2y^3 & x^3y^3 \end{bmatrix} \tag{4.19}$$

que formando una combinación lineal de sus términos produce el polinomio bicúbico siguiente

$$g_h(x,y) = a_1 + a_2x + a_3x^2 + a_4x^3 + a_5y + a_6xy + a_7x^2y + a_8x^3y + a_9y^2 + a_{10}xy^2 + a_{11}x^2y^2 + a_{12}x^3y^2 + a_{13}y^3 + a_{14}xy^3 + a_{15}x^2y^3 + a_{16}x^3y^3 \tag{4.20}$$

con $(x,y) \in \Omega_i$. De lo anterior se pueden determinar las 16 constantes a partir del sistema

$$\begin{bmatrix}
 1 & x_1 & x_1^2 & \cdot & \cdot & \cdot & x_1 y_1 & x_1^2 y_1 & \cdot & \cdot & \cdot & x_1^3 y_1^3 & a_1 \\
 1 & x_2 & x_2^2 & \cdot & \cdot & \cdot & x_2 y_1 & x_2^2 y_1 & \cdot & \cdot & \cdot & x_2^3 y_1^3 & a_2 \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 1 & x_3 & x_3^2 & \cdot & \cdot & \cdot & x_3 y_2 & x_3^2 y_2 & \cdot & \cdot & \cdot & x_3^3 y_2^3 & a_7 \\
 1 & x_4 & x_4^2 & \cdot & \cdot & \cdot & x_4 y_2 & x_4^2 y_2 & \cdot & \cdot & \cdot & x_4^3 y_2^3 & a_8 \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 1 & x_4 & x_4^2 & \cdot & \cdot & \cdot & x_4 y_4 & x_4^2 y_4 & \cdot & \cdot & \cdot & x_4^3 y_4^3 & a_{16}
 \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ \cdot \\ \cdot \\ b_{32} \\ b_{42} \\ \cdot \\ \cdot \\ b_{44} \end{bmatrix} \tag{4.21}$$

El problema a resolver es un sistema de ecuaciones lineales de la forma

$$Ax = b, \tag{4.22}$$

donde A es una matriz de 16×16 y corresponde a las evaluaciones del polinomio bicúbico propuesto, en cada uno de los puntos de la malla de elemento finito; x ⁽³⁾ es un vector de 16×1 y corresponde a las 16 incógnitas que es necesario encontrar y b es el vector lado derecho que corresponde al valor de cada parámetro del Banco original. En la Figura 4.5 se muestra el marco de referencia empleado para el sistema anterior.

Resumiendo, se requieren 16 Bancos de entrada del mismo tipo, para resolver un problema de interpolación bicúbica para que, una vez obtenido el polinomio interpolante, simplemente se evalúen en los valores de x y y (Gd_2O_3 y ^{235}U , respectivamente) proporcionados por el usuario para encontrar un nuevo Banco.

Es importante hacer la siguiente observación, cuando se inicia el proceso para generar un nuevo Banco, se requiere un conjunto de 16 de éstos, obtenidos por medio del proceso convencional⁴; lo anterior para generar una malla en X y Y , esto es, en las variables de uranio y gadolinia respectivamente; durante todo el proceso

³ Aunque en el sistema 4.21 a es el vector de incógnitas, a partir de este momento lo denotaremos por x , debido a que es la notación más común.

⁴ Se refiere a los pasos descritos en el Capítulo 1, con los códigos respectivos.

la malla creada no se modifica, y por la forma en que se genera el sistema de ecuaciones lineales a resolver (ver 4.21), la matriz es la misma para todos los parámetros que se interpolan, esto es, lo único que cambia en el sistema, es el vector lado derecho del mismo, que precisamente son los valores de cada uno de los parámetros del Banco, en cada uno de los puntos de la malla. Concluyendo, por cada proceso en el cual se empleen Bancos diferentes para generar uno nuevo, solamente se utiliza una matriz, por lo tanto, únicamente se tiene que verificar una vez el valor del número de condición respectivo, esta característica es muy importante en la solución del problema, pues con ello existe un ahorro importante de recursos.

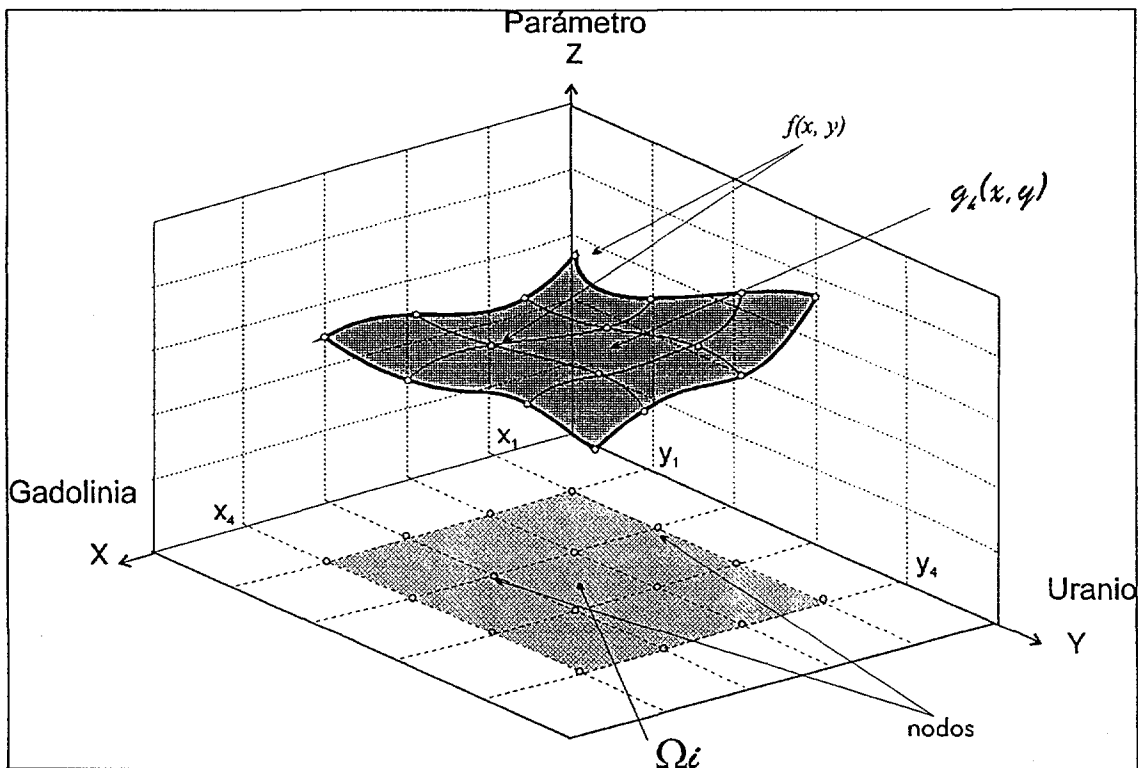


Figura 4.5. Marco de referencia para el sistema obtenido con elemento finito.

Existen diferentes métodos para resolver el sistema 4.22, cada uno con sus bondades y sus fallas; a continuación se presentan las dos propuestas empleadas para el desarrollo de las dos versiones del código *INTPOLBI*, de igual manera se harán las observaciones pertinentes con respecto al número de condición de la matriz obtenida, debido a que, de las versiones mencionadas, una de las diferencias notables fueron los resultados obtenidos para dicho valor.

4.3.- Solución Empleando Factorización LU

La primera versión del código *INTPOLBI* emplea el método de eliminación Gaussiana con pivoteo parcial. A continuación se presenta el desarrollo matemático del mismo, haciendo hincapié en algunas de las partes del método, con la idea de que en la sección posterior se tenga más claro cuales fueron los beneficios de las modificaciones hechas a esta versión.

El problema a resolver es encontrar un x tal que satisfaga 4.22, donde A es una matriz 16×16 ⁽⁵⁾ que representa el conjunto de ecuaciones lineales del sistema, x es el vector solución y b es el vector lado derecho del sistema en cuestión, estos últimos son de tamaño 16×1 , de acuerdo al sistema obtenido en el proceso.

De entrada se supone que A es no singular, sin embargo, más adelante se analizará esta situación haciendo mención al número de condición de la matriz anterior.

El método empleado para resolver el sistema es el de eliminación Gaussiana con pivoteo parcial, esto es, se involucran dos pasos durante el proceso: el primero se refiere a la eliminación hacia adelante, mientras que el segundo involucra la sustitución hacia atrás. La idea es encontrar L y U matrices 16×16 tal que

$$A = LU, \quad 4.23$$

donde L es triangular inferior unitaria y U es triangular superior. La matriz U se obtiene de la manera siguiente:

$$U = M_{n-1}P_{n-1}M_{n-2}P_{n-2} \dots M_2P_2M_1P_1A, \quad 4.24$$

donde las P_i son matrices de permutación empleadas como intercambio en el pivoteo parcial y las M_i son transformaciones de Gauss de la forma

$$M_k = I - \tau_k e_k^T \quad 4.25$$

⁵ No se olvide que en el caso de los Bancos, los sistemas resultantes son de esta dimensión.

donde

$$\tau_k^T = (\underbrace{0, \dots, 0}_k, \tau_{k+1}, \dots, \tau_n) \quad \tau_i = a_{i,k}/a_{k,k}, \quad i = k+1, \dots, n. \quad 4.26$$

La matriz L se obtiene de la manera siguiente

$$L = M_1^{-1} \dots M_{n-1}^{-1}. \quad 4.27$$

La solución final del sistema original se obtiene resolviendo el sistema

$$Ux = y, \quad 4.28$$

donde

$$y = M_{n-1}P_{n-1}M_{n-2}P_{n-2} \dots M_2P_2M_1P_1b, \quad 4.29$$

el algoritmo requiere $2n^3/3$ flops.

Por otra parte, es necesario hacer algunas observaciones acerca del método; primero: es necesario aclarar que la x^* calculada al aplicar el método anterior satisface lo siguiente [7]

$$(A + E)x^* = b, \quad 4.30$$

donde

$$|E| \leq nu(3|A| + 5P^T|L||U|) + O(u^2) \quad 4.31$$

y

$$u = \beta^t, \quad 4.32$$

aquí β es la base con t dígitos, empleada en la aritmética de punto flotante. Ahora, el pivoteo implica que los elementos de L están acotados por uno, de esta manera se tiene que

$$\|L\|_{\infty} \leq n, \quad 4.33$$

con lo cual obtenemos la cota

$$\|E\|_{\infty} \leq nu(3\|A\|_{\infty} + 5n\|U\|_{\infty}) + O(u^2). \quad 4.34$$

Si se define el factor de crecimiento como

$$\rho = \max_{i,j,k} \frac{a_{ij}^{(k)}}{\|A\|_{\infty}}, \quad 4.35$$

donde $A^{(k)}$ se calcula como $A^{(k)} = M_k P_k \dots M_1 P_1 A$, de donde se sigue que

$$\|E\|_{\infty} \leq 8n^3 \rho \|A\|_{\infty} u + O(u^2), \quad 4.36$$

de la relación anterior, el factor de crecimiento indica qué tan grande son los números durante el proceso de eliminación, por último, se puede resumir que 4.35 está acotando el error.

Se sabe que muy rara vez se encuentra la solución exacta de un sistema de ecuaciones lineales, debido a los errores de redondeo que surgen de manera natural al implementar algún algoritmo en la computadora⁶, del análisis anterior se desprende que el método de eliminación Gaussiana debe usarse con las debidas precauciones.

⁶ De hecho, en algunos casos desde el momento de almacenar las cantidades con las cuales se trabaja existe el problema citado.

4.3.1.- Número de Condición y Factorización LU

De los comentarios y el análisis anteriores se puede hacer la siguiente pregunta: ¿cuánto se altera la solución del sistema, si existen perturbaciones en los coeficientes del mismo?, en otras palabras, cómo se puede medir la sensibilidad en la solución x^* al cambiar A y b .

Veamos el siguiente análisis para responder la pregunta anterior, inicialmente se tiene

$$Ax = b. \quad 4.37$$

Si x^* es la solución obtenida, entonces

$$A(x-x^*) = b - Ax^*, \quad 4.38$$

si A es no singular, entonces

$$x - x^* = A^{-1}(b - Ax^*), \quad 4.39$$

que no es otra cosa que el error expresado en términos de la inversa, por otro lado de 4.30 se tiene que

$$\|b - Ax^*\| = \|Ex^*\| \leq \|E\| \|x^*\|. \quad 4.40$$

Sustituyendo 4.40 en 4.39 se tiene que

$$\|x - x^*\| \leq \|A^{-1}\| \|E\| \|x^*\|, \quad 4.41$$

si se define

$$\frac{\|E\|}{\|A\|} = \psi \beta^{-1}. \quad 4.42$$

En general ψ será más pequeño que β , sustituyendo 4.42 en 4.41 y dividiendo entre la norma de x^* se tiene que

$$\frac{\|x - x^*\|}{\|x^*\|} \leq \psi \|A\| \|A^{-1}\| \beta^{-t}, \quad 4.43$$

donde, el factor

$$\|A\| \|A^{-1}\|, \quad 4.44$$

se le conoce como el número de condición para el problema de inversión de la matriz A ($k_\infty(A)$), con lo cual se tiene una forma de acotar el error para la solución encontrada, empleando la definición del condicional de una matriz; el análisis anterior tiene una gran importancia debido a que, por una parte se tiene conocimiento de la sensibilidad de x^* con respecto a la solución real; por otro lado, se involucra la inversa de la matriz A . En general es fácil calcular la norma de A , sin embargo, no se puede decir lo mismo con respecto a la norma de la inversa, de hecho, podría consumir el triple de tiempo que el requerido para todo el proceso de eliminación Gaussiana, aún más, si la matriz es singular⁷, el valor obtenido para dicho parámetro simplemente será erróneo, por tal motivo, es necesario encontrar una aproximación al número de condición, para darle la vuelta al problema del cálculo de la inversa de la matriz.

Existen diferentes formas para encontrar la aproximación a dicho valor, en el caso de la primera versión del código *INTPOLBI*, la aproximación se hace de la siguiente manera⁸. Si a_j , $j=1, \dots, n$, son las columnas de A y a_j^* , $j=1, \dots, n$, son las columnas de A^{-1} , entonces en términos de la norma⁹ de un vector tenemos que

$$k_\infty(A) = \max_j \|a_j\| \cdot \max_j \|a_j^*\| \quad 4.45$$

⁷ De hecho, se debe cuidar la singularidad no sólo desde el punto de vista analítico, sino también desde el punto de vista numérico.

⁸ El método al que se hace referencia se incluye en las rutinas *DECOMP* y *SOLVE*, los comentarios acerca de éstas se harán en el siguiente capítulo.

⁹ En esta parte nos referiremos a la norma euclidiana, debido a que es el tipo de norma empleada en las rutinas *DECOMP* y *SOLVE* para los cálculos aquí señalados.

entonces

$$k_{\infty}(A) \approx \max_j \left\| a_j \right\| \frac{\|z\|}{\|y\|}, \quad 4.46$$

donde y y z son dos vectores calculados de tal manera que cumplen [8]

$$\frac{\|z\|}{\|y\|} \approx \|A^{-1}\|. \quad 4.47$$

Lo anterior involucra la solución de los sistemas de ecuaciones

$$A^T y = e \quad \text{y} \quad Az = y, \quad 4.48$$

donde e es un vector con componentes ± 1 , elegido de tal manera que maximiza el incremento durante la sustitución hacia atrás para y . Es necesario aclarar que ésta es solamente una cota inferior del número de condición real de la matriz, sin embargo, se han hecho pruebas para aceptarlo como una buena aproximación [8].

4.4.- Solución Empleando Diferencias Divididas

En esta sección se desarrolla el camino alternativo para la solución de los sistemas de ecuaciones, que surgen al aplicar la interpolación bicúbica, y el cual se implementó para obtener la segunda versión del código *INTPOLBI*.

En la Sección 4.2 se explicó la forma de obtener el sistema de ecuaciones al aplicar el método de elemento finito a un Banco, particularmente en 4.18 se propuso el producto tensorial aplicado a nuestro problema, posteriormente en 4.19 se mostró la matriz obtenida y finalmente en 4.20 se propuso el polinomio empleado; en todo el desarrollo mencionado se utilizó la base canónica, sin embargo, el sistema resultante (4.21) es inestable debido a la estructura que tiene la matriz A ⁽¹⁰⁾, en ese sentido la propuesta para mejorar los resultados es la siguiente.

¹⁰ La matriz que se obtiene es similar a una matriz de Vandermonde.

Los sistemas de ecuaciones a resolver surgen de la discretización del dominio, basándose en los valores de uranio y gadolinia que se proporcionan inicialmente, en la Figura 4.6 se muestra dicha partición con estos valores, observando desde el eje Z hacia el plano XY.

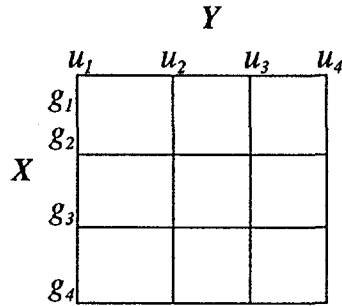


Figura 4.6. Discretización del dominio, vista en el plano XY.

Donde los u_i 's y g_i 's corresponden a los valores de uranio y gadolinia respectivamente. Empleando la base de Newton, proponemos el siguiente producto tensorial

$$\begin{bmatrix} 1 \\ y-u_1 \\ (y-u_1)(y-u_2) \\ (y-u_1)(y-u_2)(y-u_3) \end{bmatrix} \begin{bmatrix} 1 & x-g_1 & (x-g_1)(x-g_2) & (x-g_1)(x-g_2)(x-g_3) \end{bmatrix} \quad 4.49$$

con lo cual se obtiene la matriz

$$A = [c_1 \ c_2 \ c_3 \ c_4] \quad 4.50$$

donde los c_i 's son vectores columna definidos como

$$c_1 = \begin{bmatrix} 1 \\ y-u_1 \\ (y-u_1)(y-u_2) \\ (y-u_1)(y-u_2)(y-u_3) \end{bmatrix} \quad 4.51$$

$$c_2 = \begin{bmatrix} x - g_1 \\ (y - u_1)(x - g_1) \\ (y - u_1)(y - u_2)(x - g_1) \\ (y - u_1)(y - u_2)(y - u_3)(x - g_1) \end{bmatrix} \quad 4.52$$

$$c_3 = \begin{bmatrix} (x - g_1)(x - g_2) \\ (y - u_1)(x - g_1)(x - g_2) \\ (y - u_1)(y - u_2)(x - g_1)(x - g_2) \\ (y - u_1)(y - u_2)(y - u_3)(x - g_1)(x - g_2) \end{bmatrix} \quad 4.53$$

$$c_4 = \begin{bmatrix} (x - g_1)(x - g_2)(x - g_3) \\ (y - u_1)(x - g_1)(x - g_2)(x - g_3) \\ (y - u_1)(y - u_2)(x - g_1)(x - g_2)(x - g_3) \\ (y - u_1)(y - u_2)(y - u_3)(x - g_1)(x - g_2)(x - g_3) \end{bmatrix} \quad 4.54$$

formando una combinación lineal de sus términos, se produce el polinomio bicúbico siguiente

$$\begin{aligned} p(x, y) = & a_1 + a_2(x - g_1) + a_3(x - g_1)(x - g_2) + a_4(x - g_1)(x - g_2)(x - g_3) + \\ & a_5(y - u_1) + a_6(x - g_1)(y - u_1) + a_7(x - g_1)(x - g_2)(y - u_1) + \\ & a_8(x - g_1)(x - g_2)(x - g_3)(y - u_1) + a_9(y - u_1)(y - u_2) + \\ & a_{10}(x - g_1)(y - u_1)(y - u_2) + a_{11}(x - g_1)(x - g_2)(y - u_1)(y - u_2) + \\ & a_{12}(x - g_1)(x - g_2)(x - g_3)(y - u_1)(y - u_2) + \\ & a_{13}(y - u_1)(y - u_2)(y - u_3) + a_{14}(x - g_1)(y - u_1)(y - u_2)(y - u_3) + \\ & a_{15}(x - g_1)(x - g_2)(y - u_1)(y - u_2)(y - u_3) + \\ & a_{16}(x - g_1)(x - g_2)(x - g_3)(y - u_1)(y - u_2)(y - u_3) \end{aligned} \quad 4.55$$

Al evaluar $p(x, y)$ en los puntos de la malla original y en el orden que se muestra a continuación, se obtiene un sistema de la forma $Ax = b$

- 1) (g_1, u_1) 2) (g_2, u_1) 3) (g_3, u_1) 4) (g_4, u_1) 5) (g_1, u_2) 6) (g_2, u_2)
 7) (g_3, u_2) 8) (g_4, u_2) 9) (g_1, u_3) 10) (g_2, u_3) 11) (g_3, u_3) 12) (g_4, u_3)
 13) (g_1, u_4) 14) (g_2, u_4) 15) (g_3, u_4) 16) (g_4, u_4)

Donde la matriz A tiene la forma

$$A = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ A_1 & (u_2 - u_1)A_1 & \mathbf{0} & \mathbf{0} \\ A_1 & (u_3 - u_1)A_1 & (u_3 - u_1)(u_3 - u_2)A_1 & \mathbf{0} \\ A_1 & (u_4 - u_1)A_1 & (u_4 - u_1)(u_4 - u_2)A_1 & (u_4 - u_1)(u_4 - u_2)(u_4 - u_3)A_1 \end{bmatrix} \quad 4.56$$

con

$$A_1 = \begin{bmatrix} 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & (g_2 - g_1) & \mathbf{0} & \mathbf{0} \\ 1 & (g_3 - g_1) & (g_3 - g_1)(g_3 - g_2) & \mathbf{0} \\ 1 & (g_4 - g_1) & (g_4 - g_1)(g_4 - g_2) & (g_4 - g_1)(g_4 - g_2)(g_4 - g_3) \end{bmatrix} \quad 4.57$$

y

$$\mathbf{0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad 4.58$$

Siendo b el vector lado derecho del sistema, que no es otra cosa que los valores de cada uno de los parámetros del Banco, y x el vector solución que se trata de encontrar.

Veamos ahora, si se realizan las siguientes operaciones elementales por filas a la matriz $A^{(1)}$

¹¹ F_1 corresponde a las cuatro primeras filas de A , F_2 a las siguientes cuatro y así sucesivamente.

- 1) $F_4 = (F_4 - F_3)/(u_4 - u_3)$
- 2) $F_3 = (F_3 - F_2)/(u_3 - u_2)$
- 3) $F_2 = (F_2 - F_1)/(u_2 - u_1)$
- 4) $F_4 = (F_4 - F_3)/(u_4 - u_2)$
- 5) $F_3 = (F_3 - F_2)/(u_3 - u_1)$
- 6) $F_4 = (F_4 - F_3)/(u_4 - u_1)$

se obtiene la matriz siguiente

$$\hat{A} = \begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 \\ 0 & 0 & A_1 & 0 \\ 0 & 0 & 0 & A_1 \end{bmatrix} \quad 4.59$$

no se debe perder de vista que el sistema original es empleando la matriz 4.56, por tal motivo, es necesario aplicar las mismas operaciones al vector lado derecho, al hacer ésto se obtiene lo siguiente

$$\hat{b} = \begin{bmatrix} b_1 \\ \frac{b_2 - b_1}{u_2 - u_1} \\ \frac{b_3 - b_1}{(u_3 - u_2)(u_3 - u_1)} - \frac{b_2 - b_1}{(u_3 - u_2)(u_2 - u_1)} \\ \frac{b_4 - b_1}{(u_4 - u_3)(u_4 - u_2)(u_4 - u_1)} - \frac{b_3 - b_1}{(u_4 - u_3)(u_3 - u_2)(u_3 - u_1)} + \frac{b_2 - b_1}{(u_4 - u_2)(u_3 - u_2)(u_2 - u_1)} \end{bmatrix} \quad 4.60$$

donde

$$b_1 = \begin{bmatrix} p_{k1} \\ p_{k2} \\ p_{k3} \\ p_{k4} \end{bmatrix} \quad 4.61$$

$$b_2 = \begin{bmatrix} p_{k5} \\ p_{k6} \\ p_{k7} \\ p_{k8} \end{bmatrix} \quad 4.62$$

$$b_3 = \begin{bmatrix} p_{k9} \\ p_{k10} \\ p_{k11} \\ p_{k12} \end{bmatrix} \quad 4.63$$

$$b_4 = \begin{bmatrix} p_{k13} \\ p_{k14} \\ p_{k15} \\ p_{k16} \end{bmatrix} \quad 4.64$$

Además, los p_{ki} , $i = 1, \dots, 16$ corresponden a los valores de los parámetros del Banco. Resumiendo, el nuevo sistema a resolver es el siguiente

$$\hat{A}x = \hat{b}. \quad 4.65$$

Donde la matriz y el vector lado derecho señalados, tienen la estructura vista anteriormente. Por otra parte, de la nueva estructura obtenida para el sistema (4.59 y 4.60), se puede apreciar que, el problema se reduce a resolver cuatro sistemas de 4×4 , donde únicamente cambia el vector lado derecho, esto es, se tiene

$$A_i x^* = b^*, \quad 4.66$$

Donde A_i es una matriz de 4×4 con la estructura vista anteriormente (4.57), b^* toma cuatro diferentes valores y se calcula de acuerdo a 4.60, y finalmente, x^* corresponde a un vector de 4×1 que, al resolver los cuatro sistemas será la solución buscada¹².

¹² La solución del primer sistema proporciona los 4 primeros valores de la solución; al resolver el segundo sistema, se obtienen los siguientes 4 coeficientes de la solución y así sucesivamente hasta obtener los 16 valores de la solución.

Del desarrollo anterior se puede proponer el siguiente algoritmo para hallar la solución del sistema 4.65, que será exactamente la misma que la del sistema original; es importante mencionar que el algoritmo siguiente se aplica tantas veces como parámetros existen en el Banco.

Algoritmo

Para $i = 1:4$

$$P_{kni} = P_{ki}$$

Para $i = 5:8$

$$P_{kni} = \frac{P_{ki} - P_{k(i-4)}}{u_2 - u_1}$$

Para $i = 9:12$

$$P_{kni} = \frac{P_{ki} - P_{k(i-4)}}{(u_3 - u_2)(u_3 - u_1)} - \frac{P_{k(i-4)} - P_{k(i-8)}}{(u_3 - u_1)(u_2 - u_1)}$$

Para $i = 13:16$

$$P_{kni} = \frac{P_{ki} - P_{k(i-4)}}{(u_4 - u_3)(u_4 - u_2)(u_4 - u_1)} - \frac{(P_{k(i-4)} - P_{k(i-8)})(u_4 + u_3 - u_2 - u_1)}{(u_4 - u_2)(u_4 - u_1)(u_3 - u_2)(u_3 - u_1)} + \frac{P_{k(i-8)} - P_{k(i-12)}}{(u_4 - u_1)(u_3 - u_1)(u_2 - u_1)}$$

Para $i = 1:4$

$$x_{(i-1)4+1} = P_{kn(i-1)4+1}$$

$$x_{(i-1)4+2} = \frac{P_{kn(i-1)4+2} - P_{kn(i-1)4+1}}{g_2 - g_1}$$

$$x_{(i-1)4+3} = \frac{P_{kn(i-1)4+3} - P_{kn(i-1)4+1}}{(g_3 - g_2)(g_3 - g_1)} - \frac{P_{kn(i-1)4+2} - P_{kn(i-1)4+1}}{(g_3 - g_2)(g_2 - g_1)}$$

$$x_{(i-1)4+4} = \frac{P_{kn(i-1)4+4} - P_{kn(i-1)4+1}}{(g_4 - g_3)(g_4 - g_2)(g_4 - g_1)} - \frac{P_{kn(i-1)4+3} - P_{kn(i-1)4+1}}{(g_4 - g_3)(g_3 - g_2)(g_3 - g_1)} + \frac{P_{kn(i-1)4+2} - P_{kn(i-1)4+1}}{(g_4 - g_2)(g_3 - g_2)(g_2 - g_1)}$$

Con el algoritmo anterior se obtiene la solución al sistema original, esto es, los coeficientes del polinomio interpolante buscado, en el entendido de que, existe una solución para cada parámetro del Banco.

Ahora veamos qué sucede con el número de condición de la matriz obtenida con el desarrollo anterior, pues como ya se había dicho, uno de los cambios sustanciales en esta nueva versión, son los resultados para el condicional de la matriz, de hecho, con la mejoría obtenida en el citado valor, la solución encontrada para los sistemas de ecuaciones, tienen buenos resultados.

4.4.1.- Número de Condición y Diferencias Divididas

Ya vimos cuál fue el resultado de aplicar las diferencias divididas a la matriz original A , de hecho, se obtiene una matriz diagonal por bloques, donde aparece la matriz A_1 (4.56), donde ésta última es triangular inferior. Veamos qué sucede al calcular el número de condición de esta nueva matriz.

Se sabe que el número de condición de una matriz se obtiene como

$$k_{\infty}(A) = \|A\|_{\infty} \|A^{-1}\|_{\infty}. \quad 4.67$$

Ahora, la nueva matriz está dada por

$$\hat{A} = \begin{bmatrix} A_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A_1 \end{bmatrix} \quad 4.68$$

donde A_1 es triangular inferior como se vió anteriormente, por lo cual se tiene que

$$\hat{A}^{-1} = \begin{bmatrix} A_1^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_1^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_1^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A_1^{-1} \end{bmatrix} \quad 4.69$$

Si se toma en cuenta la norma infinito para matrices, claramente se puede observar que

$$\|\hat{A}\|_{\infty} = \|A_1\|_{\infty}, \quad 4.70$$

y que

$$\|\hat{A}^{-1}\|_{\infty} = \|A_1^{-1}\|_{\infty}. \quad 4.71$$

Por otro lado, ya se mencionó que A_1 es triangular inferior, por tal motivo se tiene que

$$\det(A_1) = \prod_{i=1}^4 a_{ii}, \quad 4.72$$

donde a_{ii} son las entradas de la diagonal de la matriz A_1 ; de la estructura de esta última (4.57) se tiene que

$$\det(A_1) = (g_2 - g_1)(g_3 - g_1)(g_3 - g_2)(g_4 - g_1)(g_4 - g_2)(g_4 - g_3) \quad 4.73$$

de acuerdo a los valores de gadolinia que se manejan y a la formación de la malla, entonces se cumple que

$$\det(A_1) \neq 0 \quad 4.74$$

lo cual implica que A_1 es no singular y por lo tanto es invertible, de hecho, se tiene que

$$A_1^{-1} = \begin{bmatrix} \frac{1}{g_2 - g_1} & \frac{0}{g_2 - g_1} & 0 & 0 \\ \frac{1}{(g_3 - g_1)(g_2 - g_1)} & \frac{1}{(g_3 - g_2)(g_2 - g_1)} & \frac{1}{(g_3 - g_2)(g_3 - g_1)} & 0 \\ \frac{1}{(g_4 - g_1)(g_3 - g_1)(g_2 - g_1)} & \frac{1}{(g_4 - g_2)(g_3 - g_2)(g_2 - g_1)} & \frac{1}{(g_4 - g_3)(g_3 - g_2)(g_3 - g_1)} & \frac{1}{(g_4 - g_3)(g_4 - g_2)(g_4 - g_1)} \end{bmatrix} \quad 4.75$$

Resumiendo, se tiene que

$$k_\infty(\hat{A}) = \|\hat{A}\|_\infty \|\hat{A}^{-1}\|_\infty = \|A_1\|_\infty \|A_1^{-1}\|_\infty \quad 4.76$$

Con el desarrollo anterior se observó que la matriz obtenida es menos sensible, tiene un condicional mucho menor y por lo tanto, los resultados son mejores. En el capítulo donde se muestran los resultados del trabajo, se incluyen algunos valores del condicional de la matriz para ambas versiones, donde se puede apreciar claramente lo arriba expresado.

Capítulo 5

Diseño del Sistema Intpolbi

Como se mencionó al inicio del presente trabajo, la idea fundamental fue en todo momento, tener una herramienta que facilitara la generación de Bancos de Datos Nucleares. En los capítulos anteriores se dieron las ideas necesarias para llevar a cabo esto, por lo cual, el trabajo final consistió en la programación de un sistema de cómputo que englobará todos los conocimientos vertidos anteriormente, incluyendo las restricciones que surgen de manera natural. Al sistema de cómputo creado se le llamó *INTPOLBI* (INTerpolación POLinomial BIcúbica), a continuación se presenta el diseño del mismo, esto es, la explicación detallada de cada una de las partes que lo conforman.

Es necesario aclarar que, en el presente capítulo se incluye el diseño para las dos versiones desarrolladas del código *INTPOLBI*, en el caso de la segunda versión, solo se mencionan los cambios realizados, debido a que la mayor parte del código es similar en ambas versiones. Por otra parte, se incluyen tanto rutinas como procedimientos, estos últimos se refieren a programas tipo lotes del sistema operativo Open VMS¹; adjunto a todo lo anterior se tienen los diagramas de flujo de la información de cada uno de los módulos que componen al sistema *INTPOLBI*. Asimismo, se hace referencia al número de condición de una matriz,

¹ El equipo en el cual se instaló la versión final del sistema trabaja con dicho sistema operativo.

puesto que se puso énfasis especial en dicho valor, para no tener problemas con los resultados obtenidos, adicionalmente, los valores obtenidos entre ambas versiones para dicho parámetro fue una de las diferencias importantes.

Para una mejor comprensión, el presente capítulo se divide en dos partes, esto es, una por cada uno de los módulos del sistema *INTPOLBI*; el primer módulo a su vez, incluye el diseño completo de la primera versión del código, y por otra parte, las modificaciones hechas al mismo para obtener la segunda versión. En ambas partes se refiere a la obtención del Banco utilizando el método de interpolación polinomial bicúbica; el segundo módulo se refiere a la obtención del Banco utilizando los coeficientes de los polinomios obtenidos con una interpolación previa, con la salvedad de que el nuevo Banco debe ser del mismo tipo; esto se hace tomando en cuenta que se puede utilizar la información obtenida con una interpolación, con lo cual se reduce el tiempo de ejecución aprovechando información de Bancos ya generados con anterioridad.

Es importante mencionar que el sistema *INTPOLBI* no suplanta la metodología utilizada actualmente para crear los Bancos, más bien es un apoyo para la misma, lo cual produce un ahorro significativo del tiempo y recursos utilizados en la generación de éstos. Con las pruebas realizadas hasta el momento se puede afirmar que el sistema tiene resultados bastante satisfactorios.

5.1.- Descripción General

La parte fundamental del sistema *INTPOLBI* es crear un nuevo Banco de Datos Nucleares a partir de un conjunto de 16 Bancos ya existentes, utilizando el método de interpolación bicúbica descrito en el capítulo anterior; los parámetros del nuevo Banco se obtienen evaluando los polinomios interpolantes obtenidos, con los respectivos porcentajes de gadolinia y uranio proporcionados.

En forma mas detallada se puede mencionar que el sistema trabaja sobre una arquitectura de 32 bits; no obstante, si se desean mejorar los resultados se puede instalar en una plataforma con arquitectura de 64 bits. El sistema está estructurado en forma modular para facilitar modificaciones futuras. En este sentido es importante resaltar que se puede hacer la instalación sobre cualquier plataforma siempre y cuando se cuente con el programa *TRANS* versión modificada, que permite la transformación de archivos de Bancos en formato binario a ASCII y viceversa, lo anterior porque la transferencia de archivos binarios de una plataforma a otra, en general no es posible.

Por otra parte, es importante mencionar que la versión final del sistema *INTPOLBI* se instaló en el equipo MicroVax 3100-80 perteneciente a la Gerencia de Ciencias Aplicadas del Instituto Nacional de Investigaciones Nucleares (ININ). La programación del sistema involucra dos lenguajes, por un lado VAX FORTRAN-77⁽²⁾, V. 5.0 para la parte del proceso en ambos módulos; por otra parte el lenguaje nativo del sistema operativo Open VMS V. 5.2b para la programación de procedimientos.

La estructura general del sistema se presenta en la Figura 5.1. Hay que mencionar que cada uno de los módulos se puede ejecutar por separado.

El diagrama mencionado es una presentación muy general de como se encuentra dividido el sistema; a continuación se desglosa la estructura de cada uno de los módulos, así como de las rutinas que los forman.

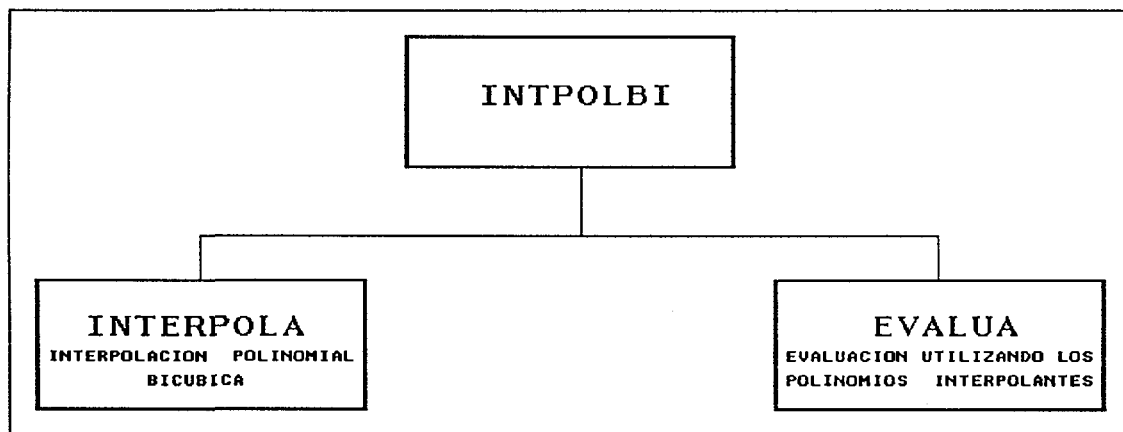


Figura 5.1. Estructura del sistema INTPOLBI.

5.2.- Estructura del Módulo Interpola 1ra. Versión

El módulo *INTERPOLA* divide su ejecución en tres partes, la primera se encarga de leer toda la información requerida (parámetros, archivos de Bancos, porcentajes de ^{235}U y Gd_2O_3 , etc.), validándola y preparándola para el proceso de interpolación. La segunda parte corresponde al proceso de interpolación en sí y por último, se escribe la información obtenida, esto es, el Banco de salida y el archivo de coeficientes de los polinomios obtenidos.

² De Digital Equipment Corporation, versión liberada en junio de 1988.

El módulo en cuestión está compuesto por una serie de rutinas en FORTRAN y algunos procedimientos en Open VMS, en la Figura 5.2 se presenta el diagrama que contiene cada una de las partes mencionadas, teniendo en cuenta que el orden en el que aparecen es el mismo que el de ejecución. Es importante señalar que cada módulo tiene asignado un procedimiento utilizado para la ejecución del mismo, éstos no se incluirán en los diagramas, por otra parte, para evitar confusiones cuando se trate de rutinas en FORTRAN no se incluirá la extensión, si esta aparece entonces se trata de un procedimiento³.

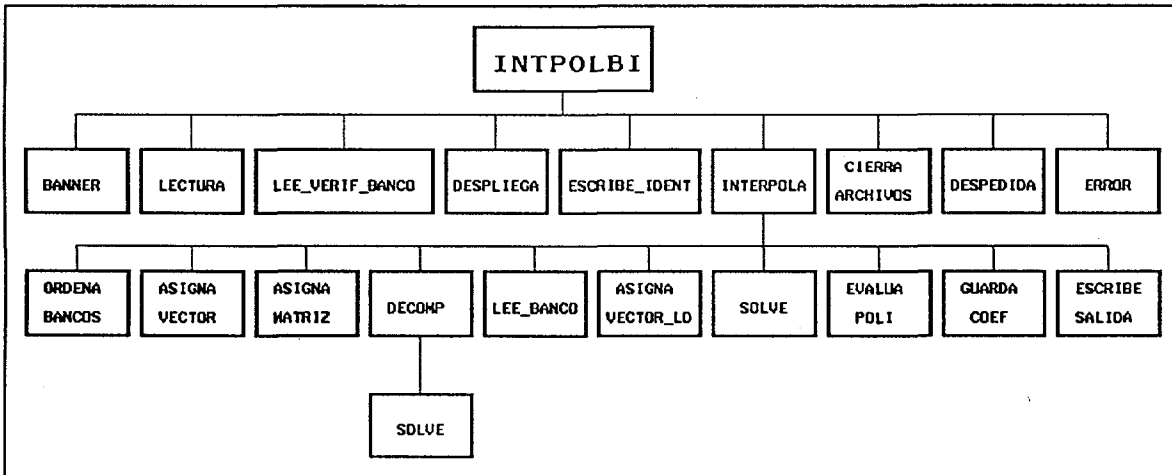


Figura 5.2. Estructura del módulo INTERPOLA.

Antes de proporcionar con más detalle la descripción de cada una de las partes que forman el módulo *INTERPOLA*, veamos el funcionamiento general del mismo con auxilio de un diagrama de flujo, en la Figura 5.3 se muestra de manera general el flujo de información, así como el proceso de interpolación en el módulo. El diagrama da una idea bastante clara de cada uno de los pasos seguidos, con lo que se puede saber de manera precisa en qué estado se encuentran los resultados en caso de una suspensión, ya sea por algún error detectado durante la ejecución o por una interrupción específica.

Es importante señalar que la formación del Banco de salida se realiza conforme se van interpolando los valores, por lo cual, en caso de que aborte la ejecución del sistema en un paso intermedio, el Banco de salida estará incompleto aunque no se marque error alguno. Por otro lado, en el diagrama también se observa que la información leída de los Bancos de entrada es por bloques, esto es, el sistema lee la información de un "caso" completo en los 16 Bancos y la procesa (incluyendo la escritura del Banco de salida); al terminar vuelve a leer otro bloque de información.

³ En este caso siempre llevarán la extensión COM.

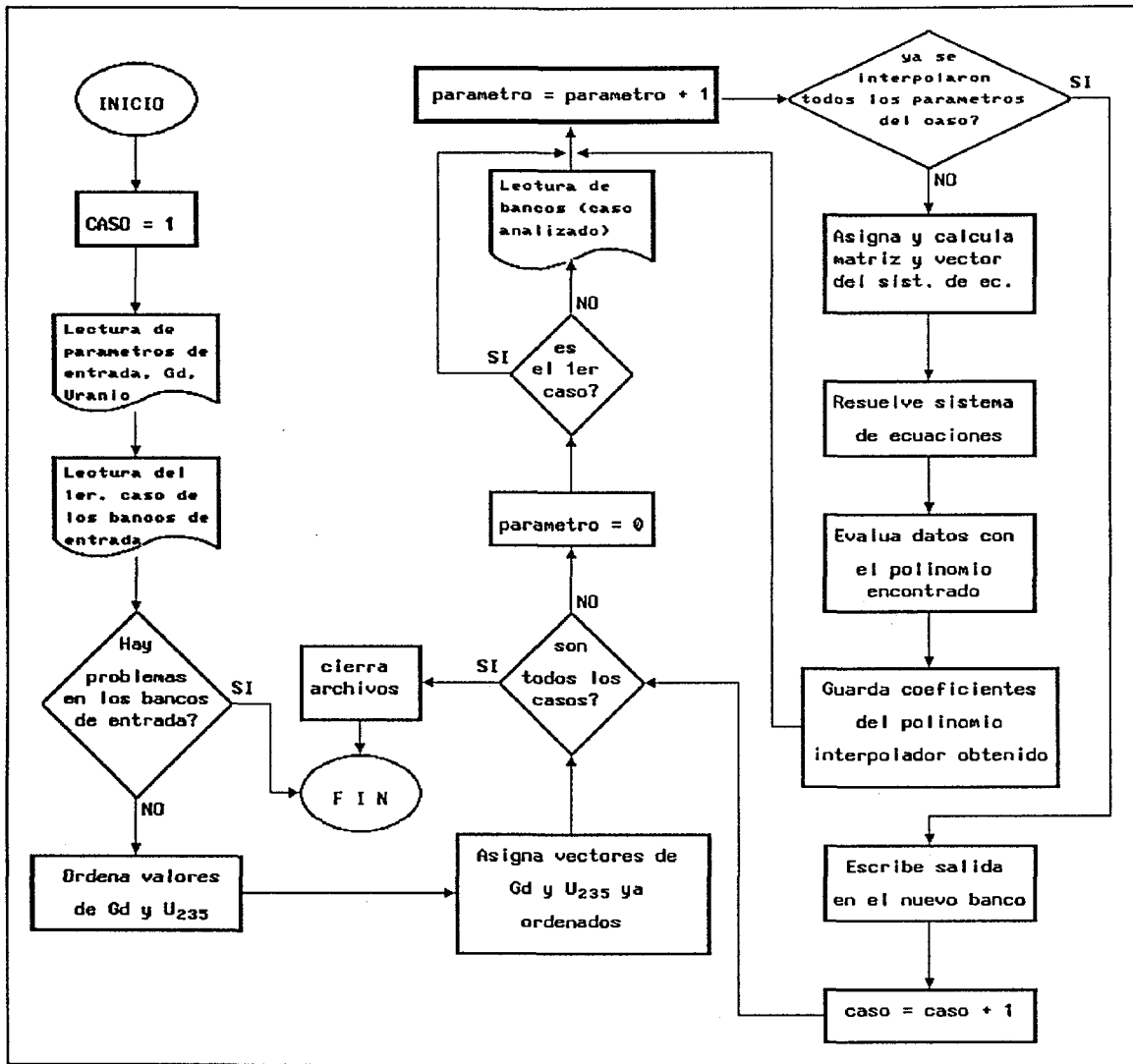


Figura 5.3. Flujo de información del módulo INTERPOLA.

A continuación se describe por separado cada una de las rutinas utilizadas en el módulo, poniendo una atención especial en aquellas que son parte medular del proceso de interpolación. En la descripción se incluye un breve resumen de la estructura, el funcionamiento, las variables que manejan y las restricciones más importantes.

INTERPOLA.COM: Es el procedimiento que se encarga de configurar el ambiente de trabajo para que sea ejecutado de manera óptima el módulo; la configuración incluye la asignación del teclado a la salida estándar, las variables globales (de la sesión) que son necesarias y el ambiente

gráfico correspondiente, dependiendo de la terminal en la cuál se ejecute. La programación de este procedimiento se hizo en Open VMS.

BANNER: Esta rutina se encarga de desplegar en la pantalla el "*banner*" que da la bienvenida al sistema; en el mismo se indica el nombre del módulo así como la versión del mismo. La rutina está programada en FORTRAN-77.

LECTURA: Esta rutina se encarga de leer los parámetros de entrada que serán utilizados para el proceso de interpolación. Dentro de los parámetros leídos están los nombres de los archivos de entrada, de salida y de coeficientes, este último contiene los coeficientes obtenidos para cada parámetro interpolado, la importancia del archivo mencionado reside en que la información contenida en éste se utiliza en el módulo de evaluación, en caso de que se desee obtener otro Banco del mismo tipo. La rutina también se encarga de leer los valores de gadolinia y uranio que sirven de base para crear el Banco de salida. Por otra parte, también es la encargada de verificar la existencia del archivo de entrada y los Bancos que se utilizarán en la interpolación. La programación de la rutina está hecha en FORTRAN-77. Algunas restricciones importantes son las siguientes: los nombres de los archivos no pueden tener más de 20 caracteres, los valores para el uranio y la gadolinia deben estar dentro de los valores permitidos.

DESPLIEGA: Esta rutina se encarga de desplegar en la pantalla los datos de entrada proporcionados por el usuario, con el fin de que sean verificados. Si los datos son correctos continúa la ejecución, en caso contrario termina el proceso. La programación de la presente rutina se hizo en FORTRAN-77.

LEE_VERIF_BANCOS: Esta rutina se encarga de dos cuestiones importantes antes del inicio de la interpolación. Primero se encarga de verificar tres cosas: la existencia de los archivos requeridos para el proceso en el directorio de trabajo, que el Banco de salida requerido no sea igual a uno de los Bancos de entrada (la verificación se hace con base en los porcentajes de uranio y gadolinia proporcionados por el usuario) y que el número de "*casos*" de los Bancos de entrada y el Banco de salida sean iguales. En segundo término, la rutina lee la primera parte de la información de los Bancos de entrada (los identificadores de las celdas y el primer "*caso*") y almacena los datos leídos en una matriz definida exprofeso para tal fin. La programación de la rutina mencionada se llevó a cabo en FORTRAN-77.

ESCRIBE IDENT: La presente rutina se encarga de escribir en el Banco de salida los identificadores de las celdas, estos se toman del último Banco proporcionado, de hecho, no existe ningún problema por tomarlos de cualquiera de los Bancos de entrada, siempre y cuando sean del mismo tipo, que es una de las restricciones más fuertes para llevar a cabo el proceso. La programación de esta rutina se hizo en FORTRAN-77.

INTERPOLA: Las funciones principales de esta rutina son: preparar la información necesaria (incluyendo formatos), coordinar el flujo de información durante el proceso, realizar la interpolación con los datos ya preparados (tomando en cuenta que el proceso se realiza por "casos") y por último escribir los resultados conforme se obtienen, en el Banco de salida. Para llevar a cabo todo lo anterior se auxilia de una serie de rutinas presentadas a continuación. La programación de todo el bloque de rutinas mencionadas se hizo en FORTRAN-77. La observación más importante de la presente rutina es la verificación que lleva a cabo con el condicional de la matriz obtenida en el proceso, el valor máximo que permite es de 1.0×10^{16} que puede ser considerado como un condicional muy alto, no obstante, el usuario lo puede modificar de acuerdo a los resultados que desee obtener. Obviamente, entre más bajo sea el condicional permitido se obtendrán mejores resultados, con la observación de que habrá más casos en los cuales el problema se aleje de la solución.

ORDENA_BANCOS: El usuario proporciona los Bancos de entrada, en general de manera aleatoria, por lo cual se hace necesario ordenarlos de acuerdo a los porcentajes de gadolinia primero y uranio después, lo anterior con el fin de formar la matriz que utilizará una de las rutinas posteriores. Todo este proceso lo lleva a cabo la presente rutina, dejando un arreglo de índices ordenados (de acuerdo a los parámetros comentados anteriormente) que servirán de apuntadores para trabajar los datos posteriormente.

ASIGNA_VECTOR: Esta rutina se encarga de asignar los valores de gadolinia y uranio de los Bancos de entrada en los vectores X y Y respectivamente, asignación que se hace con los valores ya ordenados.

ASIGNA_MATRIZ: La presente rutina se encarga de crear la matriz utilizada para la interpolación, con base en los valores de gadolinia y uranio de los Bancos de entrada. La matriz resultante es cuadrada de orden 16 y se

obtiene con base en la descripción del método de interpolación polinomial bicúbica analizado en capítulos anteriores.

DECOMP: Esta rutina se tomó de la referencia [8] y se encarga de obtener la descomposición LU de la matriz original, para ello utiliza el método de eliminación Gaussiana con pivoteo parcial, descrito en el capítulo anterior, de forma detallada. Al concluir el proceso, la rutina guarda los multiplicadores y la información del pivoteo en dos vectores previamente definidos, ésta información se utiliza posteriormente para la solución del sistema. Por otra parte, la rutina se encarga de calcular una estimación del número de condición para la matriz original, dicha estimación es una cota inferior del valor real, calculada de acuerdo al procedimiento descrito en el capítulo anterior. Al finalizar el proceso, la matriz original sale de la rutina conteniendo la siguiente información: una matriz triangular superior U y una versión permutada de la matriz triangular inferior $I - L$, tal que $P*A = L*U$.

SOLVE: Al igual que la rutina anterior, ésta también se tomó de la referencia [8], de hecho, es la parte complementaria de *DECOMP*, en este caso, la presente rutina hace uso de la información obtenida por *DECOMP* para obtener la solución del sistema, esto es, lleva a cabo la eliminación hacia delante primero, y posteriormente la sustitución hacia atrás. Es importante mencionar que, es necesario verificar el estimador del número de condición para la matriz original, obtenido por *DECOMP*, puesto que, si éste resulta muy grande, existe la posibilidad de una división por cero por parte de la presente rutina. Al concluir el proceso, la solución final se encuentra almacenada en el vector empleado para proporcionar el lado derecho del sistema.

LEE_BANCO: En este caso, la rutina mencionada se encarga de leer la información de cada "caso" de los Bancos de entrada. Es importante señalar que la presente rutina no lee el primer "caso" de los Bancos de entrada, debido a que para verificar la igualdad de los Bancos de entrada con el de salida (rutina *LEE_VERIF_BANCOS*), es necesario tener la información del primer "caso".

ASIGNA_VECTOR_LD: Una vez que se hizo la lectura de los Bancos correspondientes, esta rutina se encarga de asignar el vector lado derecho para la solución del sistema, lo cual lleva a cabo tomando el

mismo parámetro de cada uno de los Bancos de entrada, es decir, forma un vector de tamaño 16. Es importante señalar que si el vector lado derecho es idénticamente cero, la solución de la interpolación será igual, por lo tanto no se requiere de ningún cálculo; en este caso, la rutina indica tal situación.

EVALUA_POLI: Una vez obtenida la solución del sistema de ecuaciones, formada con los valores de los Bancos de entrada y los datos del Banco de salida -ya se mencionó que la solución de dicho sistema no es otra cosa que los coeficientes del polinomio interpolante para el parámetro analizado en ese momento- el paso a seguir es evaluar el polinomio obtenido con los valores de gadolinia y uranio del Banco de salida. El proceso anterior lo lleva a cabo esta rutina que, al final de la evaluación, coloca el resultado en el arreglo definido exprofeso (de hecho es el mismo arreglo que el utilizado para los Bancos de entrada, colocando los resultados en la última columna).

GUARDA_COEF: Esta rutina se encarga de guardar en un archivo los coeficientes de los polinomios interpolantes obtenidos para cada uno de los parámetros procesados; al final se tendrán tantos juegos de coeficientes, como valores se hayan interpolado. El archivo obtenido está en formato binario para evitar cualquier modificación.

ESCRIBE_SALIDA: Como se mencionó al principio la interpolación se realiza por partes, esto es, se lee todo un "caso", se interpola y finalmente se escribe en el Banco de salida. Este proceso lo lleva a cabo la presente rutina, tomando en cuenta que los valores que no son modificados los toma del último Banco de entrada proporcionado.

CIERRA_ARCHIVOS: Esta rutina se encarga de cerrar los archivos binarios utilizados durante el proceso, con el fin de evitar un posible error en caso de que alguno de ellos quedara abierto al finalizar la ejecución.

DESPEDIDA: La presente rutina se encarga de mostrar en pantalla el mensaje final, indicando con ello la finalización del proceso si no hubo problemas. Asimismo, en este mensaje se incluyen los nombres de los archivos de salida, esto es, el nombre del Banco generado y el archivo de coeficientes.

ERROR: Es la base de datos utilizada para mensajes de error, es decir, cada vez que el sistema detecta algún error durante el proceso, le transfiere a la rutina el código de error respectivo con lo cual se despliega en pantalla el mensaje adecuado. En general, una vez presentado el mensaje de error respectivo se concluye la ejecución del sistema.

5.3.- Modificaciones para la 2da. Versión

Ya se ha comentado a lo largo del presente trabajo, el desarrollo implementado para conseguir una segunda versión del código *INTPOLBI*, con la idea de obtener mejores resultados, tanto numéricos como en tiempo de CPU, en ese sentido, en esta sección se muestran las modificaciones hechas al código en cuestión, y con las cuales se obtuvo lo antes mencionado. Como la mayor parte del código es similar en ambas versiones, aquí simplemente mencionaremos aquellas partes donde se llevaron a cabo modificaciones, en el entendido de que, aquella parte que no se mencione, es la misma en ambas versiones, y por lo tanto, se puede revisar la sección anterior para aclarar cualquier duda al respecto.

En términos generales, podemos dividir los cambios realizados en tres grupos bien definidos, el primero de ellos abarca aquellas rutinas que sufrieron cambios menores, esto es, la parte medular del proceso es la misma; en un segundo grupo se pueden colocar aquellas rutinas cuyos cambios son de consideración; finalmente, en el tercer grupo se incluyen las rutinas que son completamente nuevas.

Para comenzar en forma ordenada, en primer lugar se describirán las rutinas con modificaciones menores, indicando en que consistió dicho cambio, para ello es necesario que se tome en cuenta la Figura 5.2 de la sección anterior, donde se describe la estructura del módulo *INTERPOLA*.

ORDENA BANCOS: En la primera versión del código, esta rutina ordena los Bancos en forma ascendente de acuerdo al porcentaje de gadolinia primero, y al de uranio después; en esta segunda versión, simplemente se invierte el parámetro de ordenación, esto es, primero se hace con respecto al uranio y después con respecto al porcentaje de gadolinia.

ASIGNA VECTOR: A diferencia de la primera versión, donde se asigna el producto cruz de la malla generada, en cuanto a los porcentajes de gadolinia e

uranio, en vectores X y Y respectivamente, cada uno de tamaño 16; en este caso, únicamente se asignan los valores de gadolinia e uranio en X y Y respectivamente, esto es, dos vectores de 4 entradas.

EVALUA_POLI: Al igual que el caso anterior, el polinomio resultante en ambas versiones es diferente, debido a que la base empleada así lo es; por tal motivo, en esta rutina simplemente se modificó la expresión empleada para la evaluación del polinomio.

Veamos ahora, en que casos fue necesario hacer modificaciones con mayor relevancia, explicando en que consistieron las mismas.

ASIGNA_MATRIZ: Debido a que la base empleada para la obtención de la matriz cambia de acuerdo a la versión, esto es, mientras que en el primer caso se emplea la base canónica, en la segunda versión se utiliza la base de Newton, por ende el cálculo de la matriz se modifica; adicionalmente, en la segunda versión, se emplearon diferencias divididas para lograr cierta estructura en la matriz, que facilitará los cálculos, con lo cual el problema se redujo a la solución de cuatro sistemas de ecuaciones lineales, donde únicamente se modifica el vector lado derecho, por tal motivo, se incluye el cálculo de la matriz A_1 y de su inversa.

ASIGNA_VECTOR_LD: Al aplicarle las diferencias divididas a la matriz obtenida, se logra que en lugar de resolver un sistema de 16×16 , solo fuera necesario resolver 4 sistemas de 4×4 , donde la matriz es triangular inferior y no cambia, simplemente se modifica el vector lado derecho; en este caso, la presente rutina se encarga de calcular los valores para el vector lado derecho de los 4 sistemas que se resolverán.

Finalmente, veamos los procesos que fue necesario modificar por completo. Es importante mencionar dos cuestiones fundamentales, antes de iniciar con el desarrollo; primero: de la versión inicial se eliminaron las rutinas *DECOMP* y *SOLVE*, que son las encargadas de resolver el sistema de ecuaciones que se obtiene, sustituyéndolas por otras que toman en cuenta el método desarrollado en el capítulo anterior; segundo: en ésta versión existe una rutina que también se llama *SOLVE*, la cual es completamente diferente a la incluida en la primera versión, simplemente como ambas resuelven un sistema de ecuaciones, se dejó el mismo nombre.

NUMCOND: Esta rutina se encarga de calcular el número de condición de la matriz empleada para resolver el sistema, utilizando la definición de norma infinito para matrices. Es importante mencionar que en este caso, el cálculo de dicho parámetro es exacto⁴, mientras que en la primera versión del código se hace una estimación del mismo.

SOLVE: Esta rutina se encarga de resolver los 4 sistemas de ecuaciones que surgen al aplicar las diferencias divididas a la matriz original, para llevar a cabo esto emplea el algoritmo mostrado en el capítulo anterior.

CALCULA_RESIDUAL: Esta rutina se encarga de calcular el residual, una vez que se encontró la solución del sistema de ecuaciones, tomando en cuenta que se calculan tantos residuales como sistemas se resuelvan; en otras palabras, calcula $b - Ax^*$, donde b es el vector lado derecho, x^* es la solución encontrada y A es la matriz de dicho sistema.

DNRM2: Esta rutina calcula la norma euclidiana en doble precisión de un vector de tamaño n .

5.4.- Estructura del Módulo Evalua

Como se mencionó al principio, existe la posibilidad de obtener un nuevo Banco utilizando la información generada por una interpolación previamente hecha. La manera de hacerlo es evaluando los polinomios obtenidos y proporcionando los porcentajes de gadolinia y uranio para el nuevo Banco⁵. Para llevar a cabo todo lo anterior el sistema *INTPOLBI* cuenta con un módulo "evaluator", para lo cual basta con sólo proporcionar uno de los Bancos utilizados en la interpolación original, los porcentajes de gadolinia y uranio del nuevo Banco y el archivo de coeficientes polinomiales (este archivo es el generado por el módulo de interpolación cada vez que se realiza una ejecución del mismo). El módulo divide su funcionamiento en tres partes; primero: la lectura de parámetros de entrada; segundo: la evaluación de los polinomios y por último la escritura del Banco de salida.

⁴ Al mencionar "cálculo exacto" no se están tomando en cuenta los errores de redondeo y/o truncamiento que son inherentes al equipo.

⁵ Obviamente, el nuevo Banco será del mismo tipo que el de los creados con los coeficientes que se emplean para este proceso.

En la Figura 5.4 se muestra la estructura del módulo de evaluación, tomando en cuenta que las rutinas aparecen en el orden en el cual se ejecutan. Análogamente al módulo anterior, en la Figura señalada no se muestra el procedimiento utilizado para la ejecución del módulo; sobre el particular se abundará más en la siguiente sección.

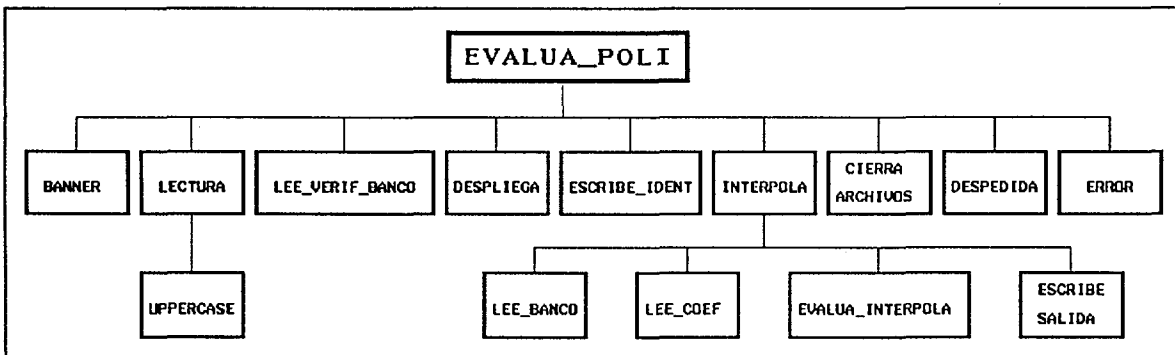


Figura 5.4. Estructura del módulo EVALUA.

Para ser consistentes en la estructura del documento, antes de discutir con más detalle cada una de las rutinas que componen al módulo *EVALUA*, en la Figura 5.5 se muestran los pasos seguidos durante el proceso de evaluación. La única observación importante en el proceso de ejecución es que, al igual que en el módulo anterior, la información se procesa en bloques, esto es, para cada "caso" se evalúan los polinomios y se escriben en el Banco de salida. Una vez concluido el proceso para un "caso", el módulo lee el siguiente bloque para procesarlo, hasta terminar todos los "casos" del Banco.

A continuación se presenta una descripción más detallada de cada una de las partes del módulo *EVALUA*; es importante hacer algunas observaciones en esta sección. Primero, es necesario aclarar que algunas de las rutinas de este módulo son las mismas que las utilizadas en el módulo anterior por lo cual, en esos casos no se incluye el nombre ni la descripción de la rutina para no ser redundantes. En segundo término, es necesario aclarar que algunas de las rutinas presentadas en la Figura 5.4 tienen el mismo nombre que las del módulo anterior, sin embargo, no realizan la misma función, en dichos casos se hará la explicación correspondiente.

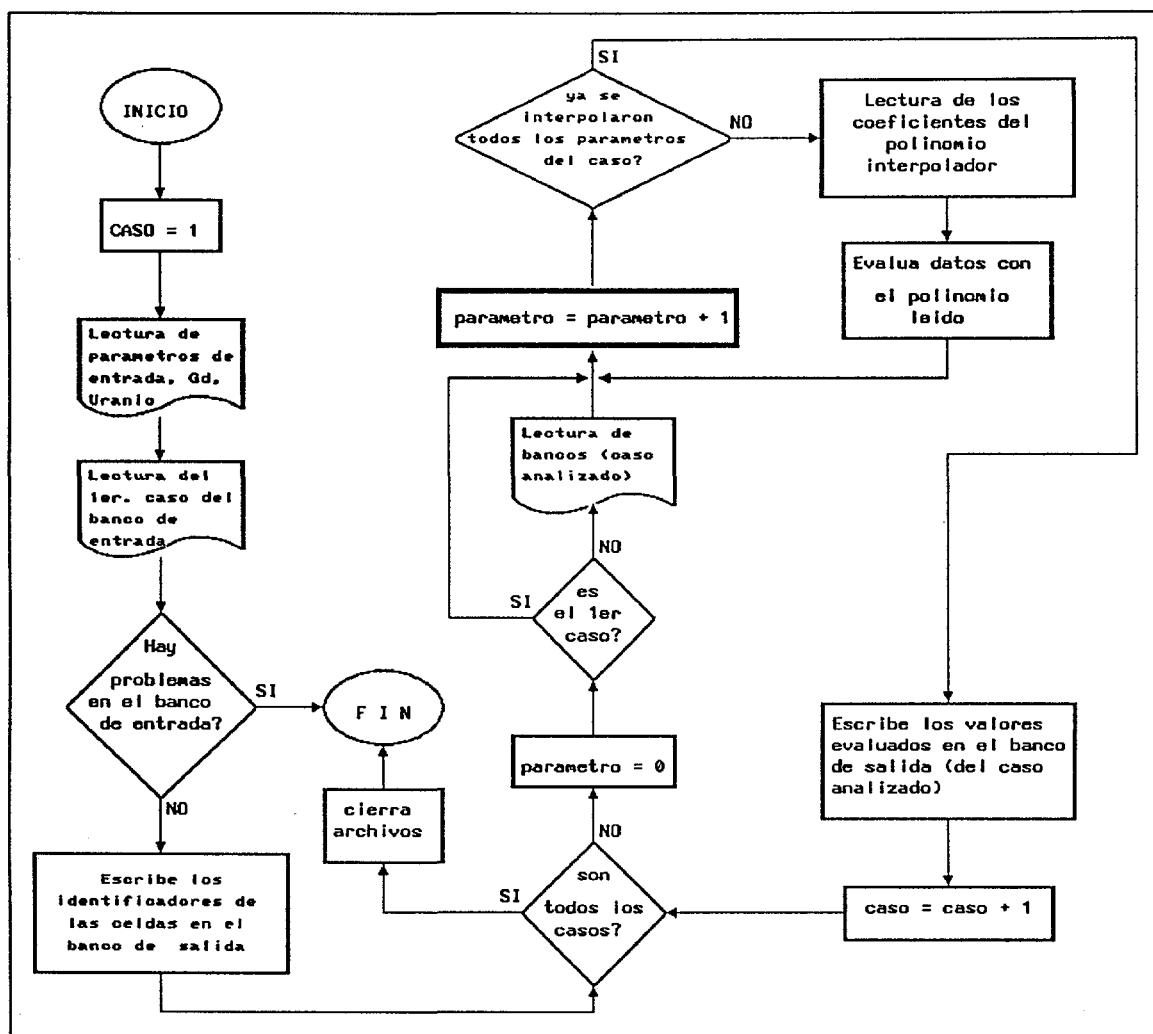


Figura 5.5. Flujo de información del módulo EVALUA.

EVALUA_INT.COM: Es el procedimiento que se encarga de configurar el ambiente de trabajo para que sea ejecutado de manera óptima el módulo. La configuración incluye la asignación del teclado a la salida estándar y las variables globales (de la sesión) que son necesarias. La programación de este procedimiento se hizo en Open VMS.

LECTURA: Esta rutina se encarga de leer los parámetros de entrada que serán utilizados para el proceso de evaluación; entre los parámetros leídos están los nombres de los archivos de entrada y salida. Dentro de los archivos de entrada están el de coeficientes (este archivo es el generado por el módulo de interpolación en una ejecución previa), el de entrada (este archivo debe ser el mismo que el utilizado en la

interpolación previa) y uno de los Bancos utilizados en la interpolación original. La rutina también se encarga de leer los valores de gadolinia y uranio que sirven de base para crear el Banco de salida; por otra parte, también se encarga de verificar la existencia de los archivos de entrada y el del Banco utilizado como base (para los valores que no se modifican) en la evaluación. La programación de la rutina está hecha en FORTRAN-77. Una restricción es la longitud en los nombres de los archivos, la cual no puede exceder de 20 caracteres.

UPPERCASE: Esta función se encarga de convertir los nombres del archivo del Banco de entrada y los nombres de los Bancos leídos en el archivo de entrada, a mayúsculas, con el objeto de no tener problemas con la verificación, ya que el sistema maneja de diferente manera mayúsculas y minúsculas.

LEE_VERIF_BANCO: Esta rutina se encarga de dos cuestiones importantes antes del inicio de la evaluación. Primero se encarga de la validación, entre estas: la existencia de los archivos requeridos para el proceso en el directorio de trabajo, que el Banco de salida requerido no sea igual al Banco de entrada (la verificación se hace en base a los porcentajes de uranio y gadolinia proporcionados por el usuario) y que el número de "casos" de los Bancos de entrada y salida sean iguales. En segundo término, la rutina lee la primera parte de la información del Banco de entrada, esto es, los identificadores de las celdas, así como los valores que no cambian en el primer "caso" y almacena los datos leídos en una matriz definida expresamente para tal fin. La programación de la rutina mencionada se llevó a cabo en FORTRAN-77.

ESCRIBE_IDENT: La presente rutina se encarga de escribir en el Banco de salida los identificadores de las celdas, los cuales se toman del Banco de entrada. La programación de esta rutina se hizo en FORTRAN-77.

INTERPOLA: Aunque tiene el mismo nombre de la rutina utilizada en el módulo anterior, en realidad su función es diferente por completo. En este caso, la rutina tiene como función principal el coordinar y preparar la información necesaria (incluyendo formatos), coordinar el flujo de información durante el proceso, realizar la evaluación con los datos ya preparados (tomando en cuenta que el proceso se realiza por "casos") y por último escribir los resultados conforme se obtienen, en el Banco de salida. Obviamente, el proceso del cual se habla ya no es una interpolación, sino una evaluación y para llevar a cabo todo lo

anterior se auxilia de una serie de rutinas presentadas a continuación. La programación de todo el bloque de rutinas mencionadas incluyendo ésta, se hizo en FORTRAN-77.

LEE_COEF: Esta rutina se encarga de leer el archivo que contiene los coeficientes de los polinomios interpolantes, con los cuales se hará la evaluación correspondiente, si los coeficientes del polinomio en turno son todos cero, entrega una bandera señalando esta situación, para que no se lleve a cabo evaluación alguna. La programación de la presente rutina se llevó a cabo en FORTRAN-77.

EVALUA_INTERPOLA: Una vez leídos los coeficientes del polinomio interpolante, correspondiente al parámetro analizado, esta rutina se encarga de evaluar este último de acuerdo a los valores de gadolinia y uranio proporcionados por el usuario, guardando el resultado en el arreglo definido expreso. Como se mencionó en la rutina anterior, en caso de que los coeficientes del polinomio en turno sean todos cero, al resultado se le asigna el valor de cero sin realizar la evaluación. La rutina está programada en FORTRAN-77.

DESPEDIDA: La presente rutina se encarga de mostrar en pantalla el mensaje final, indicando con ello la finalización del proceso si no hubo problemas. Asimismo, en el mensaje final se incluye el nombre del Banco de salida.

5.5.- Conclusiones

Con respecto al módulo *INTERPOLA* del sistema se puede mencionar lo siguiente: de acuerdo a las pruebas realizadas hasta el momento, se puede afirmar que los resultados obtenidos si bien no son excelentes, sí son una muy buena aproximación a los Bancos originales, reduciendo con ello los costos invertidos en la generación de los mismos, aún más, los resultados obtenidos con la segunda versión son mejores, debido a que el método empleado para obtener la solución de los sistemas de ecuaciones que surgen, es más eficiente y de mayor precisión. Antes de crear la versión final, se puede pensar en la posibilidad de utilizar los Bancos interpolados en cálculos de prueba o menores para los códigos *PRESTO*, de análisis de transitorios, etc.

Por otra parte, puede surgir la duda con respecto al método de eliminación Gaussiana empleado en la primera versión, para la solución de los sistemas de ecuaciones lineales que surgen durante el proceso, pues se sabe que en la actualidad existen otros métodos con mayor estabilidad en la obtención de la solución que el mencionado, de hecho, esta fue una de las razones de mayor peso para seguir en la búsqueda de un método más eficiente, el cual se consiguió al obtener la segunda versión del código. Con lo anterior, y de acuerdo a las pruebas realizadas, en general se puede afirmar que con las modificaciones realizadas, los resultados del código son buenos, de hecho, los números de condición obtenidos mejoran considerablemente al comparar ambas versiones.

Con respecto al módulo *EVALUA*, el cual vale la pena comentar que en un principio no se tenía contemplado dentro del plan de trabajo, se puede concluir lo siguiente: las dos principales razones por las cuales se programó el módulo son: en primer término, aprovechar la información generada por el módulo de interpolación, si se toma en cuenta que se desea generar una serie de Bancos del mismo tipo; en segundo lugar, el ahorro en el tiempo de ejecución, debido a que el módulo *INTERPOLA* encuentra tantos polinomios como valores se tengan en el Banco, mientras que en el caso de este módulo solamente se realizan las operaciones de evaluación. Tomando como premisas las razones expuestas anteriormente, podemos concluir que el módulo resulta bastante útil para generar Bancos de Datos Nucleares con base en información generada previamente, con lo cual en términos generales concluimos que el sistema queda completo al añadir esta parte.

Sin redundar más en las conclusiones, simplemente podemos mencionar la versatilidad del sistema, con base en los tiempos de ejecución y a los resultados obtenidos hasta el momento. Posteriormente se pueden mejorar los resultados haciendo un análisis más profundo de los diferentes métodos de interpolación que se conocen, implementando uno con mejores aproximaciones que las obtenidas hasta el momento.

Capítulo 6

Resultados

En el presente capítulo se muestran los resultados obtenidos al emplear el sistema *INTPOLBI* (con ambas versiones) en la generación de Bancos de Datos Nucleares, el análisis de éstos se hace sobre cuatro vías perfectamente definidas; la primera de ellas se basa en los resultados del sistema de acuerdo a uno de los parámetros Nucleares más importantes en la generación de un Banco, a saber, la k_{inf}^1 (k infinita) cuyo valor proporciona un buen punto de partida en la validación de los Bancos generados; en segundo término, se muestran los valores obtenidos con el código *INTPOLBI* tomando como base algunos de los parámetros más importantes y mencionados en capítulos anteriores; como tercer rubro, se verificará el tiempo de CPU empleado por el código, haciendo hincapié entre los recursos empleados al resolver los sistemas de ecuaciones y por otro lado, la evaluación de los polinomios obtenidos con una interpolación previa; finalmente, se mostrarán algunos resultados para el número de condición de la matriz que se trabaja, obtenidos con las dos versiones de *INTPOLBI*.

¹ Es necesario señalar que en este caso, la validación de dicho parámetro no es la única vía para verificar la generación de un Banco, sin embargo, si es de las más importantes debido al significado físico que dicho parámetro tiene.

De lo anterior, es necesario aclarar que, en los tres primeros casos, la comparación que se hace incluye los resultados obtenidos con el código *INTERTEG*, este último creado en el Instituto de Investigaciones Eléctricas (IIE) con el mismo propósito, como ya se mencionó en los primeros capítulos. Por otra parte, el último rubro únicamente se refiere a una de las diferencias entre las dos versiones del código *INTPOLBI*, que de hecho, fue la más importante en cuanto a repercusión en los resultados finales.

Sobre el primer punto, la forma de validar el valor del parámetro k_{inf} , se hizo generando Bancos con valores en uranio y gadolinia iguales a otros ya existentes empleando los códigos *INTPOLBI* en sus dos versiones, y el creado en el IIE cuyo nombre es *INTERTEG*, con la idea de comparar los valores obtenidos del parámetro mencionado. La comparación final se llevó a cabo calculando los errores relativos entre el valor real y el obtenido con ambos sistemas.

La segunda validación del código *INTPOLBI*, se hizo calculando los errores relativos de algunos parámetros entre un Banco real y los valores obtenidos al emplear el código. Adicionalmente y como una forma más de validar el código, se crearon tablas con los errores relativos de todos los parámetros de los Bancos generados, finalizando con un análisis de las mismas.

En el tercer caso, se calculó el tiempo de CPU consumido para obtener un nuevo Banco, empleando ambos códigos, con la consideración de que en el caso de *INTPOLBI* se calculó el tiempo de dos formas, la primera de ellas se hizo obteniendo el Banco por medio del proceso de interpolación, en sus dos versiones; en el segundo caso dicho cálculo se llevó a cabo empleando el módulo *EVALUA* del sistema; el cual simplemente evalúa los polinomios obtenidos con una interpolación previa, tomando en cuenta que el Banco generado sea del mismo tipo que alguno de los empleados anteriormente.

Con respecto al número de condición de la matriz, es necesario recordar que para la primera versión de *INTPOLBI*, la matriz obtenida es inestable, razón por la cual se buscó un método alternativo. En ese sentido, las matrices obtenidas en la segunda versión del código son menos sensibles y por lo tanto se obtienen mejores resultados; dicha sensibilidad tiene mucho que ver con el número de condición de dichas matrices, es por eso que se incluyen en este capítulo los valores obtenidos para dicho parámetro en algunos ejemplos.

Conviene aclarar que debido a la falta de espacio, no se muestran todos las comparaciones hechas durante las pruebas de validación al código creado, incluyendo las gráficas respectivas, en este caso sólo se incluye una muestra de los resultados completos, tratando en todo momento de evitar una posible parcialidad

en la comparación de los mismos. Por otra parte, las gráficas, tablas y comparaciones que se incluyen pertenecen al mismo tipo de Bancos.

6.1.- Comparación para k_{inf}

Ya se ha mencionado que un Banco de Datos Nucleares incluye diferentes parámetros, algunos de ellos se mencionaron en capítulos anteriores. De forma particular, en el capítulo 1 se mencionó la importancia que tiene el parámetro k_{inf} (k infinita) al momento de validar la generación de un Banco. La comparación del parámetro mencionado se llevó a cabo siguiendo los pasos explicados a continuación.

Paso 1) Se eligió un conjunto de Bancos ya existentes, de un total de 44, del tipo GE9B de General Electric [9], con celdas de 8x8, 8 barras con gadolinia, canal central de agua y 6 "casos". Los Bancos se encuentran en un intervalo de 4 a 32 % de gadolinia (Gd_2O_3) y de 3 a 4 % de uranio (^{235}U), del conjunto de Bancos se eligieron 16 de estos para generar la malla requerida por el código *INTPOLBI* para el proceso, la Tabla 6.1 muestra los porcentajes empleados tanto en uranio como en gadolinia para construir dicha partición.

% de Gadolinia	% de Uranio
4.0	3.0
12.8	3.3
22.4	3.6
32.0	4.0

Tabla 6.1. Porcentajes de Gd_2O_3 y ^{235}U empleados para generar la malla.

Paso 2) Se obtuvieron diez Bancos con cada una de las dos versiones del sistema *INTPOLBI*, tomando como base los 16 Bancos mencionados anteriormente. Los porcentajes de gadolinia y uranio de los Bancos obtenidos fueron los mismos en ambas versiones, dichos valores se

muestran en la Tabla 6.2, tomando en cuenta que, los nombres de los Bancos generados con la primera versión tienen el patrón "***REFCI*.REC**", mientras que para la segunda versión el patrón en dichos nombres fue "***REFCJ*.REC**".

Banco interpolado	% de Gadolinia	% de Uranio
L1REFCI1.REC L1REFCJ1.REC	22.4	3.5
L1REFCI2.REC L1REFCJ2.REC	12.8	3.2
L1REFCI3.REC L1REFCJ3.REC	32.0	3.8
L1REFCI4.REC L1REFCJ4.REC	4.0	3.9
L1REFCI5.REC L1REFCJ5.REC	32.0	3.1
L1REFCI6.REC L1REFCJ6.REC	12.8	3.7
L1REFCI7.REC L1REFCJ7.REC	4.0	3.4
L1REFCI8.REC L1REFCJ8.REC	32.0	3.4
L1REFCI9.REC L1REFCJ9.REC	4.0	3.1
L1REFCI10.REC L1REFCJ10.REC	22.4	3.9

Tabla 6.2. Porcentajes de Gd_2O_3 y ^{235}U de los Bancos obtenidos con INTPOLBI (ambas versiones), durante las pruebas.

Conviene aclarar que para la generación de los Bancos de la Tabla 6.2, se tomaron en cuenta valores de uranio y gadolinia iguales a algunos del grupo original de Bancos, con la finalidad de poder comparar los resultados con datos reales.

Paso 3) Para completar la información necesaria para llevar a cabo la comparación entre ambos códigos, se obtuvieron 20 Bancos empleando *INTERTEG*, divididos en dos grupos de diez elementos cada uno, con la idea de comparar si los resultados obtenidos con el código anterior cambiaban al

emplear un número diferente de Bancos de entrada. Las Tablas 6.3 y 6.4 muestran los porcentajes de gadolinia y uranio empleados, tanto para los Bancos de entrada, como para los Bancos obtenidos durante la interpolación².

Banco Interpolado	Valores de salida		Valores de entrada		
	(%) Gd ₂ O ₃	(%) ²³⁵ U	Bancos de entrada	Parámetro empleado	Porcentajes del parámetro, empleados en la interpolación ³
L1REFCG1.REC	22.4	3.5	2	Gadolinia	12.8, 32
L1REFCG2.REC	12.8	3.2	2	Gadolinia	4, 22.4
L1REFCG3.REC	32	3.8	2	Uranio	3.7, 3.9
L1REFCG4.REC	4	3.9	2	Uranio	3, 4
L1REFCG5.REC	32	3.1	2	Uranio	3, 3.6
L1REFCG6.REC	12.8	3.7	2	Gadolinia	4, 32
L1REFCG7.REC	4	3.4	2	Uranio	3.3, 3.5
L1REFCG8.REC	32	3.4	2	Uranio	3, 3.9
L1REFCG9.REC	4	3.1	2	Uranio	3, 3.4
L1REFCG10.REC	22.4	3.9	2	Gadolinia	12.8, 32

Tabla 6.3. Porcentajes de Gd₂O₃ y ²³⁵U de los Bancos obtenidos con INTERTEG.

² La razón por la cual se incluyen los porcentajes de uranio y gadolinia para los Bancos de entrada, es porque en cada metodología, el número de Bancos empleados para la generación es diferente, por lo tanto se manejaron intervalos similares para dichas variables.

³ Se refieren a los porcentajes empleados en el parámetro indicado en la columna anterior, en los Bancos de entrada.

Banco Interpolado	Valores de salida		Valores de entrada		
	(%) Gd_2O_3	(%) ^{235}U	Bancos de entrada	Parámetro empleado	Porcentajes del parámetro, empleados en la interpolación ⁴
L1REFCD1.REC	22.4	3.5	3	Gadolinia	4, 12.8, 32
L1REFCD2.REC	12.8	3.2	3	Gadolinia	4, 22.4, 32
L1REFCD3.REC	32	3.8	5	Uranio	3.5,3.6,3.7,3.9,4
L1REFCD4.REC	4	3.9	5	Uranio	3,3.3,3.6,3.8,4
L1REFCD5.REC	32	3.1	5	Uranio	3,3.2,3.3,3.4,3.5
L1REFCD6.REC	12.8	3.7	3	Gadolinia	4, 22.4, 32
L1REFCD7.REC	4	3.4	5	Uranio	3,3.2,3.5,3.7,3.9
L1REFCD8.REC	32	3.4	8	Uranio	3, 3.1, 3.2, 3.3, 3.5, 3.6, 3.7, 3.8
L1REFCD9.REC	4	3.1	8	Uranio	3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4
L1REFCD10.REC	22.4	3.9	3	Gadolinio	4, 12.8, 32

Tabla 6.4. Porcentajes de Gd_2O_3 y ^{235}U de los Bancos obtenidos con INTERTEG.

⁴ Se refieren a los porcentajes empleados en el parámetro indicado en la columna anterior, en los Bancos de entrada.

Paso 4) Se calcularon los errores relativos para k_{inf} entre el valor real y el calculado con cada uno de los códigos, lo anterior para los "casos"⁵ 1, 2, 4 y 5 de cada uno de los diferentes Bancos, de éstos se tomó una muestra representativa, debido a la falta de espacio.

Paso 5) Para concluir esta parte, se graficaron los errores obtenidos con respecto a los diferentes quemados con los que se trabaja en un Banco. Las gráficas respectivas corresponden a las Figuras 6.1 a 6.10. Es necesario aclarar que al generar un Banco a partir del "caso" 4 únicamente se consideran los primeros catorce pasos de quemado, por tal motivo, para los "casos" 4 y 5 se toman dichos valores. Por otra parte, al graficar los errores relativos, en el eje de las abscisas se modificó la escala debido a que en el rango de quemado de 0 a 10000 se concentra el mayor número de puntos, dando como resultado que en dichas secciones no se podía distinguir con claridad los símbolos graficados. Asimismo, los valores en X están divididos entre mil para facilitar el graficado. Finalmente, como los valores obtenidos para k_{inf} en las dos versiones de INTPOLBI son muy cercanos, sólo se grafican los valores obtenidos con la primera versión y posteriormente se adiciona un listado (Tablas 6.5 a 6.14) por gráfica con los valores obtenidos con ambas versiones, para tener más claro los resultados obtenidos.

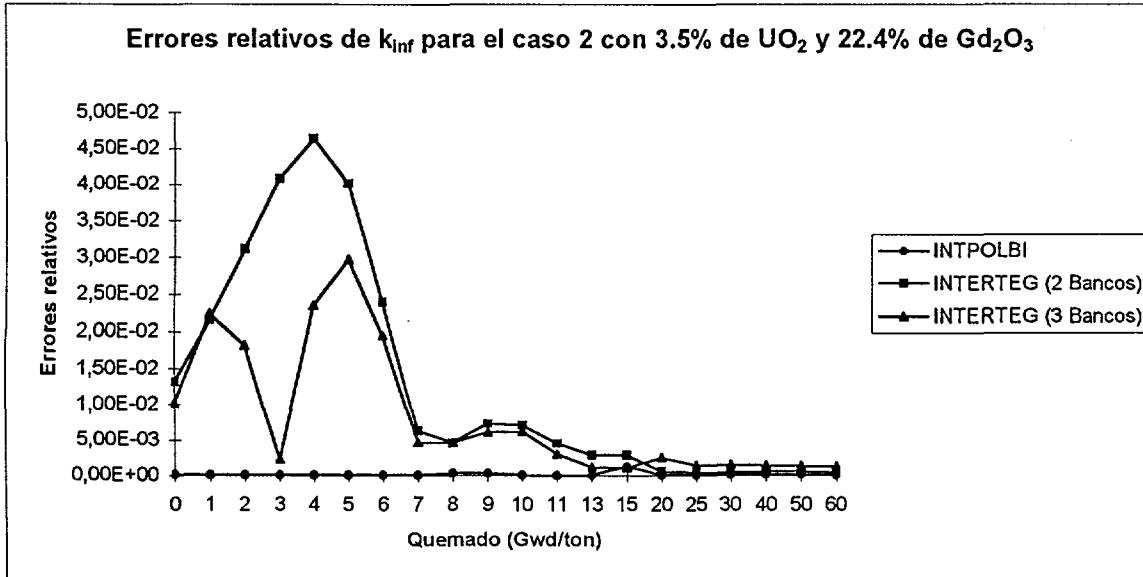


Figura 6.1. Errores para k_{inf} caso 2, Bancos L1REFC*1.REC.

⁵ Los casos 1 y 2 son sin barra de control, mientras que los casos 4 y 5 son con barra de control; por otra parte, los casos 2 y 5 tienen 40% de vacíos, mientras que el 1 y 4 tienen 0% de vacíos.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.162555E-03	0.17854634E-03
1000	0.114854E-03	0.12899900E-03
2000	0.122432E-03	0.13539538E-03
3000	0.113021E-03	0.12571155E-03
4000	0.960223E-04	0.10525317E-03
5000	0.513320E-04	0.54443011E-04
6000	0.462118E-04	0.48137299E-04
7000	0.766911E-04	0.74005380E-04
8000	0.347495E-03	0.35891763E-03
9000	0.312742E-03	0.33065811E-03
10000	0.506108E-04	0.62000734E-04
11000	0.363391E-04	0.45324035E-04
12500	0.784260E-04	0.88950277E-04
15000	0.126401E-02	0.12773708E-02
20000	0.105509E-03	0.11837059E-03
25000	0.115541E-03	0.12938328E-03
30000	0.145436E-03	0.16039421E-03
40000	0.150666E-03	0.16648385E-03
50000	0.137920E-03	0.15202921E-03
60000	0.106408E-03	0.11709512E-03

Tabla 6.5. Errores para k_{inf} caso 2, Bancos L1REFC11 y L1REFC11.

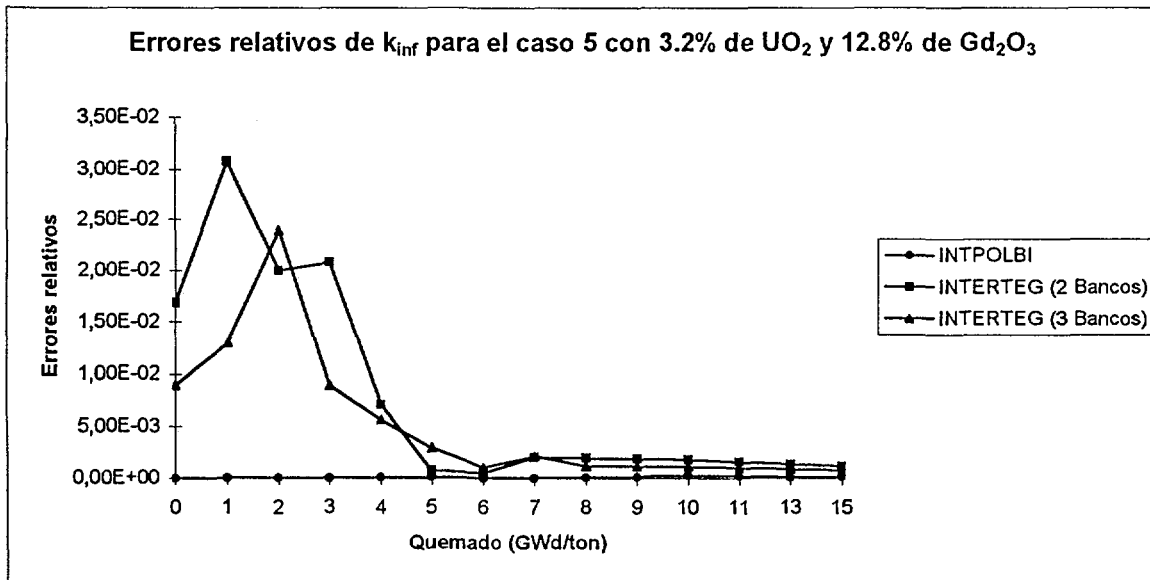


Figura 6.2. Errores para k_{inf} caso 5, Bancos L1REFC*2.REC.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.736823E-05	0.14736465E-05
1000	0.314225E-04	0.26572510E-04
2000	0.828248E-04	0.79641635E-04
3000	0.774679E-04	0.71741153E-04
4000	0.946063E-04	0.89939953E-04
5000	0.176801E-03	0.16862518E-03
6000	0.499060E-04	0.56412598E-04
7000	0.436254E-04	0.53121210E-04
8000	0.122324E-03	0.11407238E-03
9000	0.182507E-03	0.17503269E-03
10000	0.303352E-03	0.29539230E-03
11000	0.220286E-03	0.21250038E-03
12500	0.932874E-04	0.99365054E-04
15000	0.188586E-03	0.19569657E-03

Tabla 6.6. Errores para k_{inf} caso 5, Bancos L1REFCI2 y L1REFCJ2.

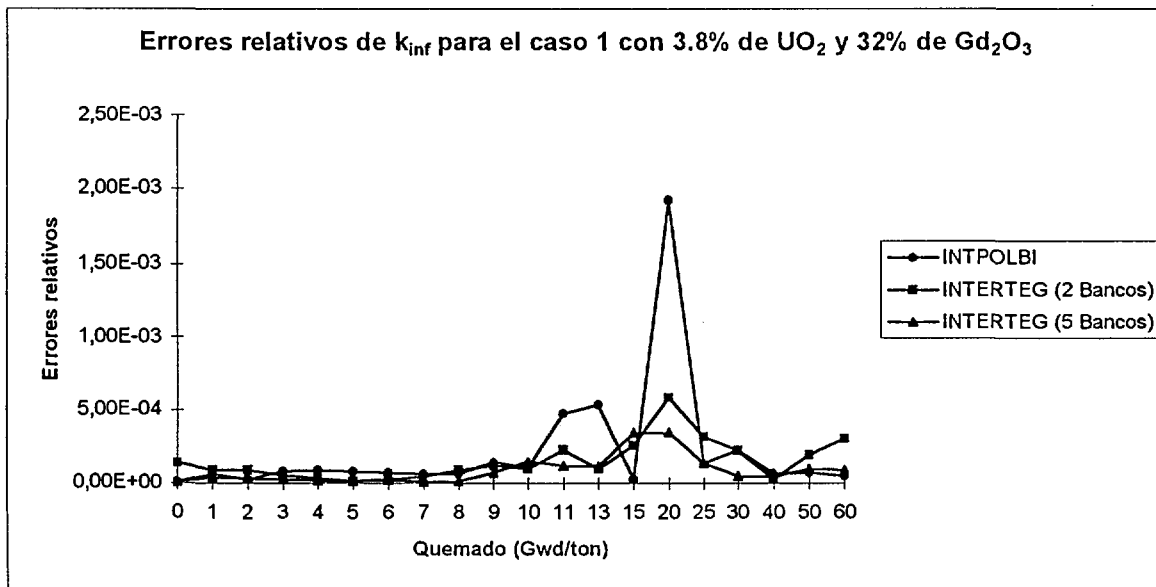


Figura 6.3. Errores para k_{inf} caso 1, Bancos L1REFC*3.REC.

Quemado	Error relativo para k_{inf} empleando <i>INTPOLBI</i>	
	Primera versión	Segunda versión
0	0.99091103E-05	0.13805951E-04
1000	0.58891022E-04	0.36419449E-04
2000	0.24559169E-04	0.38373701E-05
3000	0.78268982E-04	0.55037966E-04
4000	0.82925544E-04	0.70035567E-04
5000	0.76253491E-04	0.58199046E-04
6000	0.68085770E-04	0.55286902E-04
7000	0.57251596E-04	0.45345751E-04
8000	0.58278278E-04	0.59196849E-04
9000	0.13833528E-03	0.12204871E-03
10000	0.10253133E-03	0.10927421E-03
11000	0.47542498E-03	0.46982249E-03
12500	0.53510303E-03	0.52104465E-03
15000	0.26594316E-04	0.63224225E-05
20000	0.19227143E-02	0.19012220E-02
25000	0.13141947E-03	0.15643048E-03
30000	0.22377467E-03	0.25186452E-03
40000	0.61442239E-04	0.93318282E-04
50000	0.68263507E-04	0.98184195E-04
60000	0.46433724E-04	0.67864676E-04

Tabla 6.7. Errores para k_{inf} caso 1, Bancos L1REFC13 y L1REFC13.

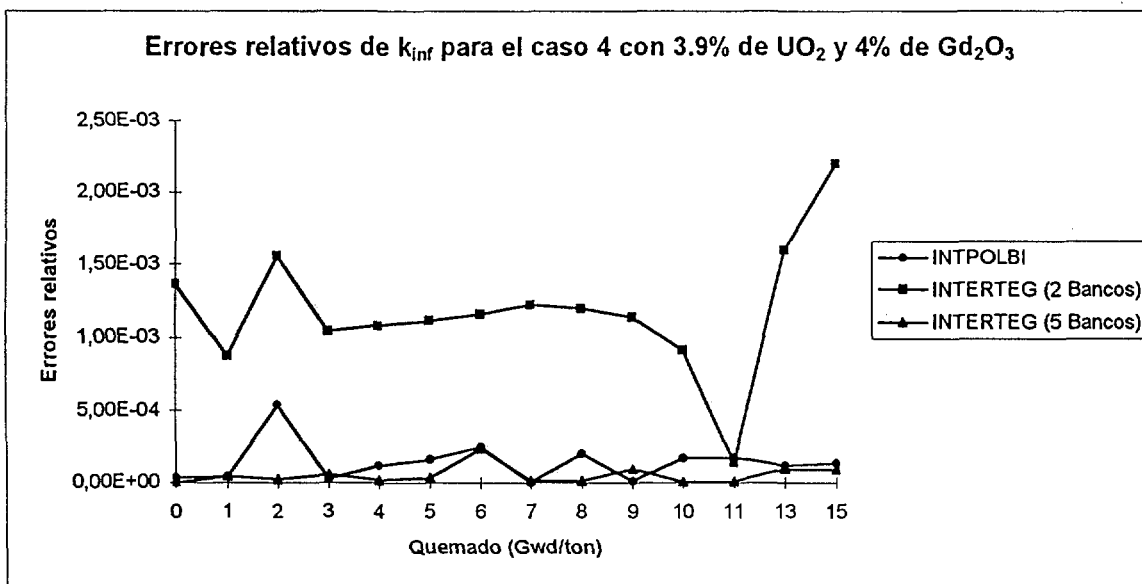


Figura 6.4. Errores para k_{inf} caso 4, Bancos L1REFC*4.REC.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.35434477E-04	0.35559027E-04
1000	0.41637417E-04	0.41524883E-04
2000	0.53890230E-03	0.53867878E-03
3000	0.26258247E-04	0.26029915E-04
4000	0.11794404E-03	0.11817645E-03
5000	0.15966446E-03	0.15978281E-03
6000	0.25055392E-03	0.25037208E-03
7000	0.22969384E-05	0.27935739E-05
8000	0.20018881E-03	0.20018881E-03
9000	0.11931118E-04	0.11931118E-04
10000	0.17131252E-03	0.17123947E-03
11000	0.17327278E-03	0.17343702E-03
12500	0.11798228E-03	0.11816691E-03
15000	0.13378363E-03	0.13388559E-03

Tabla 6.8. Errores para k_{inf} , caso 4, Bancos L1REFC14 y L1REFC14.

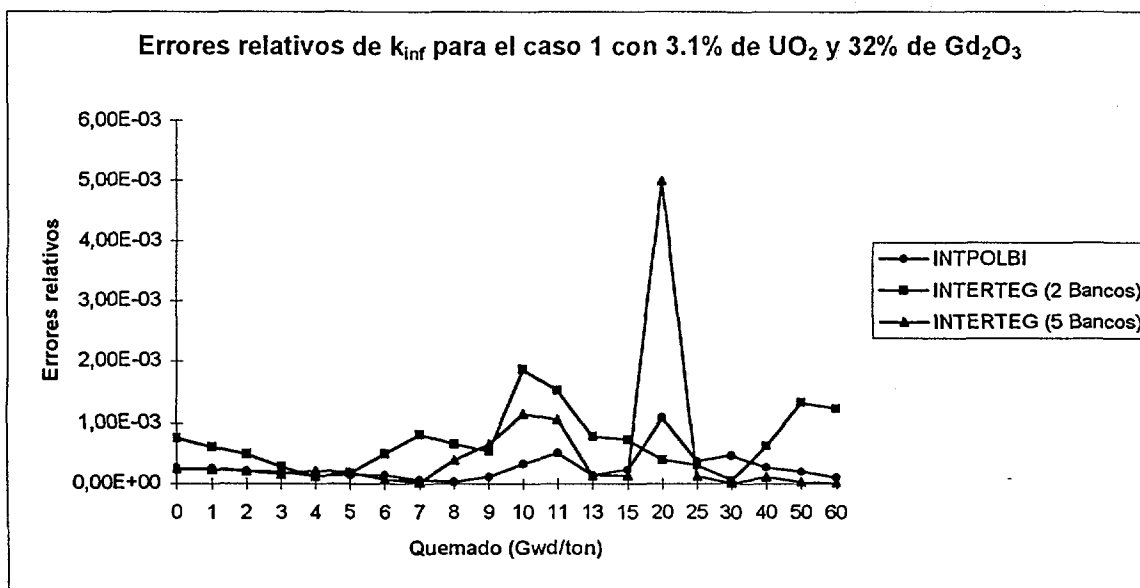


Figura 6.5. Errores para k_{inf} , caso 1, Bancos L1REFC*5.REC.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.26200869E-03	0.23597499E-03
1000	0.25014486E-03	0.22604021E-03
2000	0.21458705E-03	0.19189483E-03
3000	0.19538475E-03	0.17385349E-03
4000	0.11729289E-03	0.10187146E-03
5000	0.14736329E-03	0.13333395E-03
6000	0.14315598E-03	0.13406671E-03
7000	0.64305583E-04	0.55632787E-04
8000	0.31614112E-04	0.33990334E-04
9000	0.11292174E-03	0.12094406E-03
10000	0.33252026E-03	0.32039557E-03
11000	0.50861714E-03	0.49184618E-03
12500	0.13894575E-03	0.15679105E-03
15000	0.22999677E-03	0.19627664E-03
20000	0.10916141E-02	0.11055121E-02
25000	0.37751294E-03	0.35176933E-03
30000	0.48249564E-03	0.45490987E-03
40000	0.27370555E-03	0.24725954E-03
50000	0.19888497E-03	0.17949507E-03
60000	0.10556517E-03	0.95051044E-04

Tabla 6.9. Errores para k_{inf} caso 1, Bancos L1REFC15 y L1REFC15.

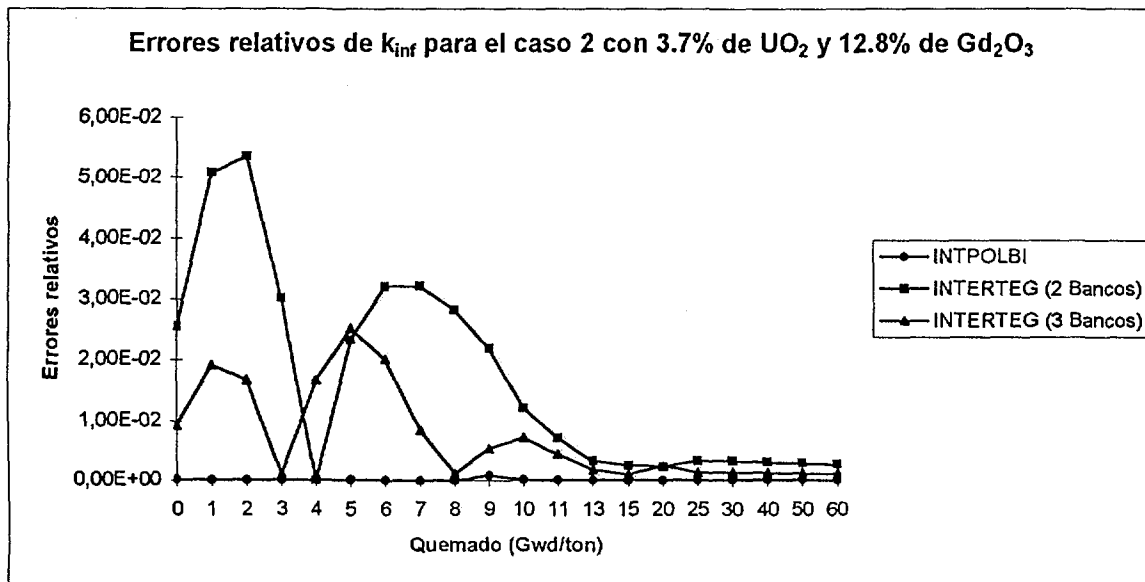


Figura 6.6. Errores para k_{inf} caso 2, Bancos L1REFC*6.REC.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.16217541E-03	0.16945180E-03
1000	0.10753501E-03	0.11365227E-03
2000	0.10065768E-03	0.10512443E-03
3000	0.73191193E-04	0.76609460E-04
4000	0.15008266E-03	0.15301016E-03
5000	0.20573818E-03	0.21043146E-03
6000	0.91742586E-04	0.95358999E-04
7000	0.20221185E-04	0.23082674E-04
8000	0.33590542E-04	0.37813355E-04
9000	0.76542539E-03	0.76358637E-03
10000	0.19766217E-03	0.19168429E-03
11000	0.10939054E-03	0.11413804E-03
12500	0.97169250E-04	0.10168175E-03
15000	0.14477861E-03	0.15606539E-03
20000	0.15235020E-03	0.15965613E-03
25000	0.13628077E-03	0.14315112E-03
30000	0.17761809E-03	0.18459045E-03
40000	0.19063693E-03	0.19853965E-03
50000	0.18187810E-03	0.18924674E-03
60000	0.12638592E-03	0.13220028E-03

Tabla 6.10. Errores para k_{inf} caso 2, Bancos L1REFC16 y L1REFC16.

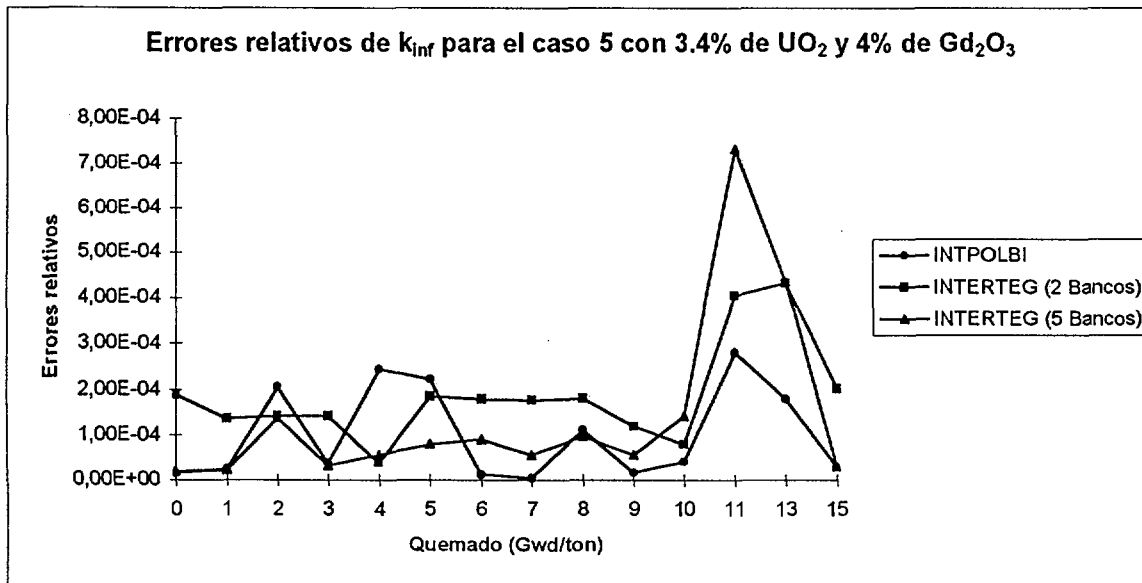


Figura 6.7. Errores para k_{inf} caso 5, Bancos L1REFC*7.REC.

Quemado	Error relativo para k_{inf} empleando <i>INTPOLBI</i>	
	Primera versión	Segunda versión
0	0.15255509E-04	0.15392328E-04
1000	0.24363824E-04	0.24424733E-04
2000	0.20487759E-03	0.20451196E-03
3000	0.38329657E-04	0.38578550E-04
4000	0.24414714E-03	0.24389356E-03
5000	0.22208593E-03	0.22208593E-03
6000	0.12378755E-04	0.12378755E-04
7000	0.41393369E-05	0.43429109E-05
8000	0.11163833E-03	0.11199500E-03
9000	0.15904305E-04	0.15904305E-04
10000	0.41241539E-04	0.41399555E-04
11000	0.28057513E-03	0.28057513E-03
12500	0.18089352E-03	0.18108373E-03
15000	0.28782826E-04	0.28680759E-04

Tabla 6.11. Errores para k_{inf} caso 5, Bancos L1REFCI7 y L1REFCJ7.

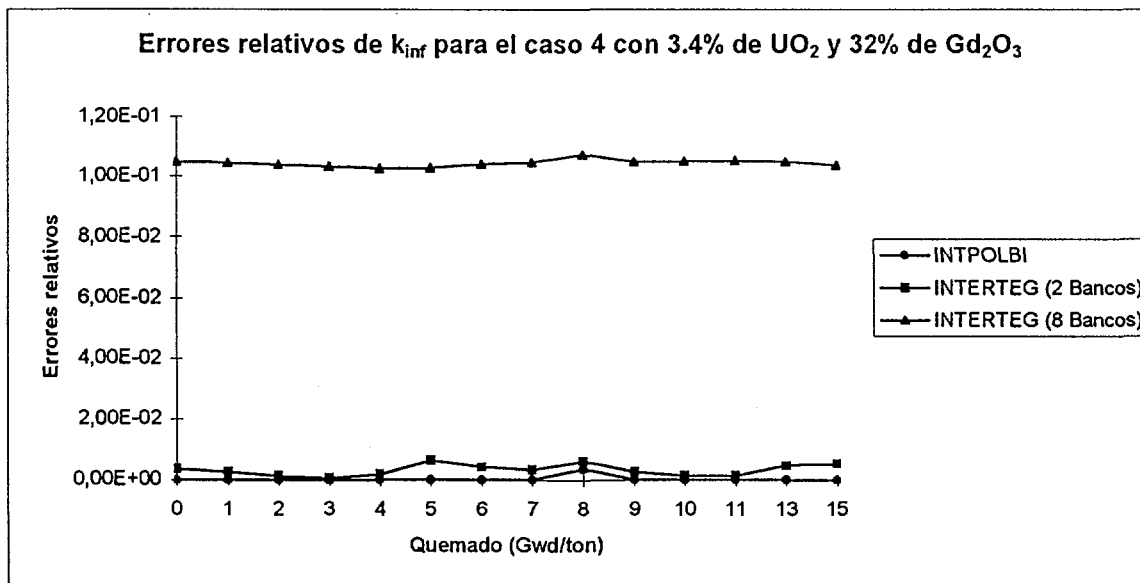


Figura 6.8. Errores para k_{inf} caso 4, Bancos L1REFC*8.REC.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.27389293E-04	0.28456407E-06
1000	0.12736925E-04	0.99528979E-05
2000	0.86101491E-05	0.27389766E-04
3000	0.25995672E-04	0.10464080E-04
4000	0.27090861E-03	0.25939816E-03
5000	0.23132616E-03	0.23646263E-03
6000	0.13731200E-03	0.15752703E-03
7000	0.20924055E-03	0.19947556E-03
8000	0.35157993E-02	0.35496280E-02
9000	0.19727084E-03	0.16843727E-03
10000	0.26651076E-03	0.23551934E-03
11000	0.15674582E-03	0.12466639E-03
12500	0.14212205E-03	0.11641396E-03
15000	0.83898769E-04	0.67877903E-04

Tabla 6.12. Errores para k_{inf} caso 4, Bancos L1REFCI8 y L1REFCJ8.

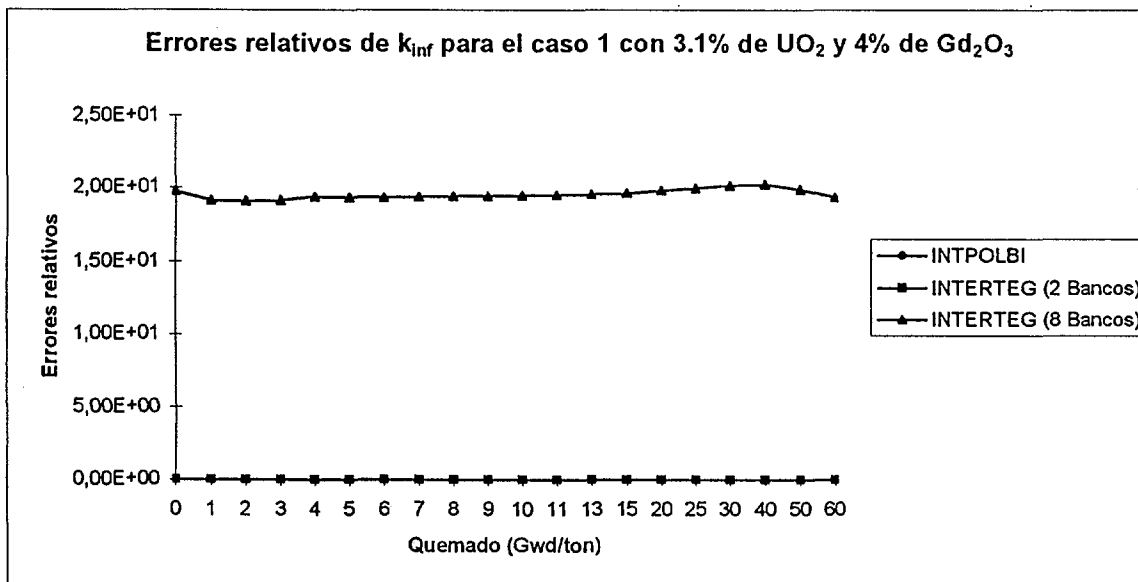


Figura 6.9. Errores para k_{inf} caso 1, Bancos L1REFC*9.REC.

Quemado	Error relativo para k_{inf} empleando INTPOLBI	
	Primera versión	Segunda versión
0	0.20908033E-03	0.20918924E-03
1000	0.13070556E-03	0.13050751E-03
2000	0.88125918E-04	0.87842556E-04
3000	0.22198526E-03	0.22142184E-03
4000	0.31198764E-02	0.31198764E-02
5000	0.16547747E-03	0.16528495E-03
6000	0.14711024E-03	0.14701301E-03
7000	0.13693445E-03	0.13673799E-03
8000	0.15554760E-03	0.15634173E-03
9000	0.15951690E-03	0.15891495E-03
10000	0.15769996E-03	0.15749714E-03
11000	0.17481696E-03	0.17471443E-03
12500	0.17107574E-03	0.17086724E-03
15000	0.20300366E-03	0.20160955E-03
20000	0.23500284E-03	0.23477555E-03
25000	0.29445771E-03	0.29445771E-03
30000	0.41816739E-03	0.41823171E-03
40000	0.21330352E-03	0.21323129E-03
50000	0.12423408E-03	0.12415444E-03
60000	0.70779512E-04	0.70864684E-04

Tabla 6.13. Errores para k_{inf} caso 1, Bancos L1REFC19 y L1REFCJ9.

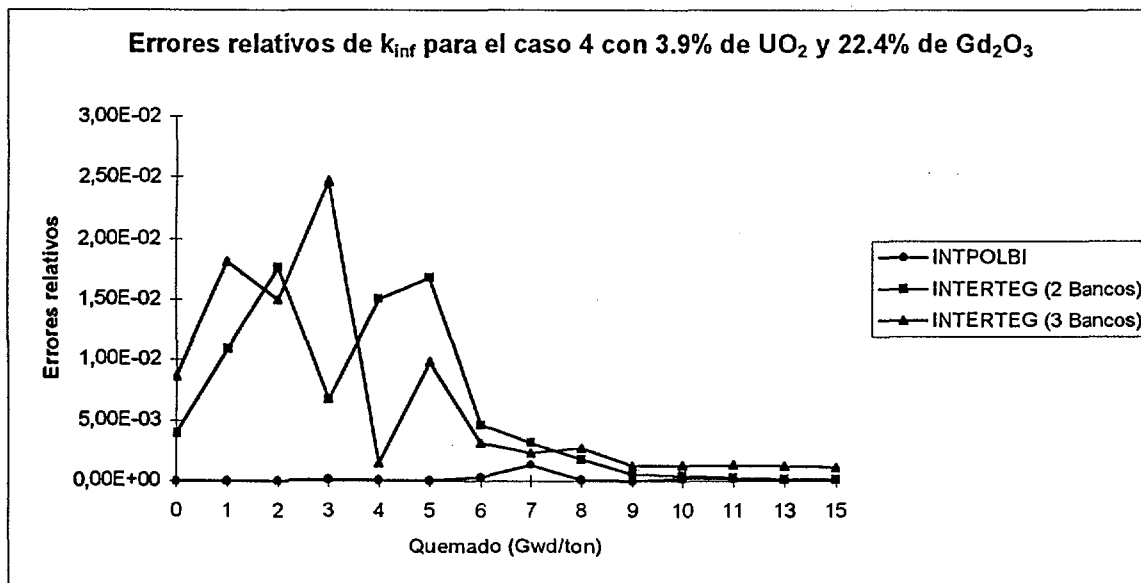


Figura 6.10. Errores para k_{inf} caso 4, Bancos L1REFC*10.REC.

Quemado	Error relativo para k_{inf} empleando <i>INTPOLBI</i>	
	Primera versión	Segunda versión
0	0.14537262E-04	0.31875283E-04
1000	0.47433277E-05	0.10201404E-04
2000	0.64234569E-05	0.37155289E-05
3000	0.13015914E-03	0.13855258E-03
4000	0.11550835E-03	0.12342150E-03
5000	0.59134989E-04	0.68558889E-04
6000	0.31453825E-03	0.30168280E-03
7000	0.13365528E-02	0.13346461E-02
8000	0.11794988E-03	0.13381524E-03
9000	0.14268710E-04	0.33407989E-04
10000	0.21041010E-03	0.23154553E-03
11000	0.19590546E-03	0.21948593E-03
12500	0.16376135E-03	0.18633014E-03
15000	0.13055143E-03	0.14728880E-03

Tabla 6.14. Errores para k_{inf} , caso 4, Bancos L1REFCI10 y L1REFCJ10.

Los comentarios que surgen a partir de las gráficas anteriores, se dejarán para la última sección del presente capítulo, con la idea de englobar los resultados y con ello tener un análisis más completo de los mismos.

6.2.- Resultados para otros Parámetros

En los primeros capítulos se mencionó la estructura que tiene un Banco de Datos Nucleares, recordemos un poco ésta para tener una idea más clara de las observaciones que se harán más adelante, con respecto a los resultados obtenidos con los códigos analizados. Un Banco está dividido en "*casos*", cada uno de los cuales consta de 878 valores, de los cuales, los primeros 38 incluyen los pasos de quemado, etiquetas, porcentajes de vacío, entre otros, y cuyos valores no se modifican, por lo tanto, en total se tienen 840 valores para cada "*caso*", que cambian dependiendo del tipo de estudio que se haga, es decir, son los datos que se interpolan durante el proceso.

Las observaciones anteriores se hicieron con la siguiente finalidad, al realizar las pruebas con el código *INTPOLBI* con ambas versiones, se calcularon los errores relativos de todos los datos interpolados, esto es, $840 \times n$ valores, donde n es el número de "casos" del Banco⁶, con la idea de verificar que los resultados fueran adecuados, particularmente en los 11 parámetros más importantes de un Banco, mencionados éstos en capítulos anteriores.

En ese sentido, las Tablas 6.15 y 6.16 presentan un pequeño resumen de los errores relativos calculados con las dos versiones del código *INTPOLBI*, incluyendo también, los resultados para el código *INTERTEG*, para ello, se tomó una cota inferior de 10^{-5} , esto es, de los 840 valores que se interpolan en cada caso, se contaron aquéllos cuyo error relativo estuviera por debajo de la cota mencionada, asimismo se muestran los errores relativos máximo y mínimo de cada caso. Conviene aclarar que la decisión de tomar la cota anterior, se hizo en base a que, para los usuarios de los Bancos una cota de 10^{-3} era suficiente para considerar que los resultados fueran buenos.

Por otra parte, en las tablas mencionadas (6.15 y 6.16) se incluye una muestra pequeña de todos los resultados obtenidos durante las pruebas realizadas, lo anterior debido a los problemas de espacio que surgen de manera natural, sin embargo, en todo momento se cuidó que los valores elegidos fueran una muestra representativa de los resultados totales.

Finalmente, conviene mencionar dos cuestiones importantes, primero: los resultados de las Tablas 6.15 y 6.16 se obtuvieron a partir del conjunto de Bancos empleados a lo largo del documento para presentar las pruebas de verificación; segundo: los comentarios acerca de las tablas y las respectivas conclusiones, se dejan para la última sección, con la idea, como ya se mencionó anteriormente, de presentar conclusiones globales.

⁶ Para los resultados aquí mostrados, se trabajaron Bancos que incluyen 6 "casos".

Código	Banco	Caso	Bancos de entrada	Errores relativos < 10 ⁻⁵	Error relativo mínimo	Error relativo máximo
INTPOLBI v1	L1REFCI1	1	-	138	0.0E0	6.2298E-3
INTPOLBI v2	L1REFCJ1	1	-	136	8.5025E-8	6.2913E-3
INTERTEG	L1REFCG1	1	2	62	7.2635E-8	2.1415E-2
INTERTEG	L1REFCD1	1	3	47	0.0E0	3.2717E-2
INTPOLBI v1	L1REFCI1	6	-	110	0.0E0	1.4393E-2
INTPOLBI v2	L1REFCJ1	6	-	103	0.0E0	1.4459E-2
INTERTEG	L1REFCG1	6	2	86	7.2635E-8	2.1415E-2
INTERTEG	L1REFCD1	6	3	79	0.0E0	3.2717E-2
INTPOLBI v1	L1REFCI2	2	-	335	0.0E0	9.0330E-3
INTPOLBI v2	L1REFCJ2	2	-	346	0.0E0	9.0631E-3
INTERTEG	L1REFCG2	2	2	40	0.0E0	5.2814E-2
INTERTEG	L1REFCD2	2	3	64	6.4622E-8	3.0407E-2
INTPOLBI v1	L1REFCI2	5	-	303	0.0E0	2.3434E-2
INTPOLBI v2	L1REFCJ2	5	-	324	0.0E0	2.3404E-2
INTERTEG	L1REFCG2	5	2	53	0.0E0	5.2814E-2
INTERTEG	L1REFCD2	5	3	76	0.0E0	3.0407E-2
INTPOLBI v1	L1REFCI3	3	-	132	0.0E0	1.4715E-2
INTPOLBI v2	L1REFCJ3	3	-	291	0.0E0	1.4784E-2
INTERTEG	L1REFCG3	3	2	154	0.0E0	2.13E-2
INTERTEG	L1REFCD3	3	5	206	0.0E0	1.6726E-2
INTPOLBI v1	L1REFCI3	4	-	223	0.0E0	1.4715E-2
INTPOLBI v2	L1REFCJ3	4	-	244	0.0E0	1.4784E-2
INTERTEG	L1REFCG3	4	2	142	0.0E0	2.13E-2
INTERTEG	L1REFCD3	4	5	197	0.0E0	1.6726E-2
INTPOLBI v1	L1REFCI4	1	-	233	0.0E0	1.1343E-2
INTPOLBI v2	L1REFCJ4	1	-	230	0.0E0	1.1345E-2
INTERTEG	L1REFCG4	1	2	39	3.9116E-7	4.7728E-2
INTERTEG	L1REFCD4	1	5	254	0.0E0	1.7254E-2
INTPOLBI v1	L1REFCI4	3	-	205	0.0E0	3.5002E-2
INTPOLBI v2	L1REFCJ4	3	-	205	0.0E0	3.4999E-2
INTERTEG	L1REFCG4	3	2	34	1.2756E-7	4.7728E-2
INTERTEG	L1REFCD4	3	5	220	0.0E0	1.7254E-2
INTPOLBI v1	L1REFCI5	2	-	113	6.1056E-8	1.7589E-2
INTPOLBI v2	L1REFCJ5	2	-	126	0.0E0	1.7680E-2
INTERTEG	L1REFCG5	2	2	34	7.0633E-8	5.2192E-2
INTERTEG	L1REFCD5	2	5	110	0.0E0	1.347E-2

Tabla 6.15. Resultados globales para los códigos analizados.

Código	Banco	Caso	Bancos de entrada	Errores relativos < 10 ⁻⁵	Error relativo mínimo	Error relativo máximo
INTPOLBI v1	L1REFCI6	6	-	116	0.0E0	3.0313E-2
INTPOLBI v2	L1REFCJ6	6	-	116	0.0E0	3.0333E-2
INTERTEG	L1REFCG6	6	2	40	6.9171E-8	9.5157E-2
INTERTEG	L1REFCD6	6	3	80	0.0E0	3.3471E-2
INTPOLBI v1	L1REFCI7	1	-	426	0.0E0	4.9653E-3
INTPOLBI v2	L1REFCJ7	1	-	425	0.0E0	4.9647E-3
INTERTEG	L1REFCG7	1	2	157	8.2512E-8	7.9993E-3
INTERTEG	L1REFCD7	1	5	304	0.0E0	7.188E-3
INTPOLBI v1	L1REFCI7	2	-	444	0.0E0	4.9653E-3
INTPOLBI v2	L1REFCJ7	2	-	443	0.0E0	4.9647E-3
INTERTEG	L1REFCG7	2	2	166	8.2512E-8	7.9993E-3
INTERTEG	L1REFCD7	2	5	292	0.0E0	7.188E-3
INTPOLBI v1	L1REFCI8	3	-	259	0.0E0	2.9497E-2
INTPOLBI v2	L1REFCJ8	3	-	345	0.0E0	2.9597E-2
INTERTEG	L1REFCG8	3	2	8	2.3542E-7	1.5184E-1
INTERTEG	L1REFCD8	3	8	0	8.1008E-2	1.4591E-1
INTPOLBI v1	L1REFCI8	4	-	223	0.0E0	2.9497E-2
INTPOLBI v2	L1REFCJ8	4	-	269	0.0E0	2.9597E-2
INTERTEG	L1REFCG8	4	2	9	2.3542E-7	1.5184E-1
INTERTEG	L1REFCD8	4	8	0	8.1008E-2	1.4591E-1
INTPOLBI v1	L1REFCI9	5	-	124	0.0E0	1.7986E-2
INTPOLBI v2	L1REFCJ9	5	-	123	0.0E0	1.7986E-2
INTERTEG	L1REFCG9	5	2	42	0.0E0	3.1427E-2
INTERTEG	L1REFCD9	5	8	0	8.9529E0	5.2287E+1
INTPOLBI v1	L1REFCI9	6	-	109	0.0E0	1.7986E-2
INTPOLBI v2	L1REFCJ9	6	-	109	0.0E0	1.7986E-2
INTERTEG	L1REFCG9	6	2	44	0.0E0	3.1427E-2
INTERTEG	L1REFCD9	6	8	0	8.9529E0	5.2287E+1
INTPOLBI v1	L1REFCI10	1	-	235	0.0E0	8.5291E-3
INTPOLBI v2	L1REFCJ10	1	-	220	0.0E0	8.5537E-3
INTERTEG	L1REFCG10	1	2	75	0.0E0	2.0077E-2
INTERTEG	L1REFCD10	1	3	76	0.0E0	2.903E-2
INTPOLBI v1	L1REFCI10	4	-	194	0.0E0	3.5034E-2
INTPOLBI v2	L1REFCJ10	4	-	193	0.0E0	3.5153E-2
INTERTEG	L1REFCG10	4	2	93	0.0E0	2.0077E-2
INTERTEG	L1REFCD10	4	3	76	0.0E0	3.5504E-2

Tabla 6.16. Resultados globales para los códigos analizados.

6.3.- Tiempos de CPU

Otro de los valores importantes que se verificaron al momento de realizar las pruebas, con los códigos mencionados a lo largo del capítulo, fue el tiempo de CPU consumido durante el proceso de interpolación, debido a que, dicho valor permite evaluar qué tan costoso puede ser un código, con respecto al tipo de métodos que emplea para resolver un problema determinado. En ese sentido, en todas las pruebas se eliminaron aquellos procesos que no tienen que ver directamente con la interpolación, con la idea de tener resultados más confiables y con ello, poder comparar entre los códigos el tiempo de CPU consumido por los procesos.

Por otra parte, conviene mencionar que, para el caso del código *INTPOLBI* se midieron tres tiempos diferentes; los dos primeros se refieren al mismo número de versiones existentes del código, y la tercera tomando en cuenta el módulo *EVALUA* del mismo, el cual, como se debe recordar, utiliza los valores de los coeficientes obtenidos con una interpolación previa, siempre y cuando se trate de obtener un Banco del mismo tipo, lo cual redundará en un ahorro en el tiempo de máquina invertido, como se podrá observar a continuación. Asimismo, en el caso del código *INTERTEG* se midieron dos tiempos diferentes, dependiendo del número de Bancos de entrada empleados para llevar a cabo la interpolación.

En la Tabla 6.17 se muestran los tiempos de CPU consumidos por los códigos para llevar a cabo el proceso de interpolación, tomando en cuenta las observaciones del párrafo anterior. Es necesario aclarar algunos detalles con respecto a la notación empleada en dicha tabla; primero: en la primera columna de la misma aparece un asterisco, debido a que, dependiendo del código se modifica una letra⁷, sin embargo, se refieren al mismo Banco obtenido, esto es, mismo tipo, porcentajes de gadolinia y uranio, y número de "casos"; segundo: los valores que aparecen entre paréntesis en las columnas que se refieren al código *INTERTEG*, indican el número de Bancos de entrada empleados para llevar a cabo la interpolación.

Finalmente, las conclusiones con respecto a la validación aquí explicada, se encuentran en la última sección del presente capítulo, lo anterior, debido a que se trata de ser consecuentes en el tratamiento que se le da a los resultados obtenidos.

⁷ Se emplearon las letras I, J, G y D, que corresponden a los códigos *INTPOLBI* primera versión, *INTPOLBI* segunda versión, *INTERTEG* con 2 Bancos de entrada e *INTERTEG* con 3 o más Bancos de entrada, respectivamente.

Banco obtenido	Tiempo de CPU consumido (segundos)				
	INTPOLBI versión 1	INTPOLBI versión 2	EVALUA	INTERTEG	INTERTEG
L1REFC*1	6.058594	6.179688	2.652344	1.421875 (2)	1.468750 (3)
L1REFC*2	6.421875	6.199219	2.042969	1.449219 (2)	1.550781 (3)
L1REFC*3	6.019531	6.148438	1.511719	1.039063 (2)	1.339844 (5)
L1REFC*4	6.031250	6.402344	1.582031	1.101563 (2)	1.800781 (5)
L1REFC*5	6.828125	6.218750	1.613281	0.976562 (2)	1.429688 (5)
L1REFC*6	6.898438	6.230469	1.531250	1.789063 (2)	1.871094 (3)
L1REFC*7	6.460938	6.269531	1.500000	1.218750 (2)	1.832031 (5)
L1REFC*8	6.691406	6.289063	1.570313	1.070313 (2)	1.796875 (8)
L1REFC*9	5.980469	6.500000	1.691406	1.191406 (2)	2.382813 (8)
L1REFC*10	6.242188	6.660156	1.820313	1.417969 (2)	1.488281 (3)

Tabla 6.17. Tiempos de CPU consumidos por los códigos.

6.4.- Número de Condición para INTPOLBI

Ya se ha mencionado a lo largo del capítulo, el trabajo realizado para obtener una segunda versión del código *INTPOLBI*, con la idea de mejorar los resultados, en dichas menciones, también se puso particular énfasis en los valores obtenidos para el número de condición de la matriz resultante al resolver el problema de la interpolación, de hecho, la mejoría en los resultados obtenidos se debió principalmente al método empleado para resolver el sistema de ecuaciones lineales resultante; recordemos que en la primera versión del código se utiliza una base canónica para construir el polinomio interpolante, dando como resultado una matriz sensible; en la segunda versión esta eventualidad se elimina al emplear una

base de Newton, lo cual permite obtener una matriz menos sensible, cuyo número de condición es mucho menor, como se podrá observar a continuación.

Adicionalmente, se debe mencionar que en la primera versión del código el número de condición calculado es en realidad una aproximación de éste, que si bien se puede considerar como una buena aproximación, no deja de serlo. Por otra parte, en la segunda versión el número de condición que se calcula es exacto, lo cual permite tener mayor confianza en los resultados.

Durante las pruebas de validación, se trabajaron diferentes conjuntos de Bancos, la mayoría del mismo tipo, lo cual no permite tener una gran cantidad de valores calculados para el número de condición, sin embargo, y a pesar de este inconveniente, en la Tabla 6.18 se muestran algunos valores obtenidos para dicho parámetro con las dos versiones del código *INTPOLBI*. De igual manera a cómo se hizo en las secciones anteriores con respecto a las conclusiones, dejaremos éstas para la última sección, remarcando la idea de concluir de forma global los resultados obtenidos.

Número de condición	
INTPOLBI v1	INTPOLBI v2
1.248275×10^{12}	2567
2.920906×10^{12}	2374
2.563215×10^{13}	2385
1.395672×10^{13}	2412

Tabla 6.18. Números de condición para las dos versiones de *INTPOLBI*.

6.5.- Conclusiones

En esta sección se harán las observaciones necesarias, para, de manera global, tener una conclusión acerca del código *INTPOLBI*, en ese sentido, dividiremos el análisis en dos grandes rubros, el primero de ellos se refiere a la comparación

hecha entre los códigos *INTPOLBI* e *INTERTEG*, el primero de éstos incluye las dos versiones creadas, mientras que para el segundo se tomará en cuenta el número de Bancos de entrada empleados para llevar a cabo la interpolación; el segundo rubro sobre el cual se harán las conclusiones globales, únicamente tomará en cuenta las dos versiones del código *INTPOLBI*, mencionando de manera más puntual en qué partes y porqué se obtuvieron mejores resultados con la segunda versión. A su vez el primer rubro se dividirá en tres partes, las cuales se desglosarán en el siguiente orden: valores para k infinita, resultados globales y finalmente tiempo de CPU empleado; es necesario señalar que el análisis no se hará de manera aislada, puesto que, la idea de no mencionar las conclusiones en cada una de las secciones anteriores, fue precisamente para tener más claro y de forma global la validación de los resultados.

Al observar las gráficas de los errores relativos para k infinita, se pueden hacer varias afirmaciones al respecto, a continuación se hará un breve análisis de acuerdo a dichas gráficas y las repercusiones que tienen con respecto a los resultados obtenidos al generar un nuevo Banco de Datos Nucleares.

Los resultados obtenidos con *INTPOLBI* son consistentes en todos los casos, esto es, independientemente de los valores que se trabajen para los porcentajes de uranio y gadolinia del nuevo Banco, los errores relativos obtenidos para el parámetro en cuestión no varían considerablemente, lo cual no sucede con el código *INTERTEG*, esto es, se puede observar que si el número de Bancos aumenta existe la posibilidad de obtener resultados erróneos como se muestra en las Figuras 6.8 y 6.9, donde se trabajaron 8 Bancos de entrada. Adicionalmente, también se puede observar que los resultados obtenidos con *INTERTEG*, varían en la mayoría de casos, dependiendo del número de Bancos de entrada que se trabajen.

Por otra parte, también es importante mencionar que entre el 97 y 99 por ciento de los errores relativos para k infinita, que se obtuvieron con *INTPOLBI*, en cualquiera de sus dos versiones, se encuentran entre 10^{-4} y 10^{-5} , lo cual indica que se tiene una muy buena aproximación para uno de los parámetros más importantes en la generación de un Banco, mientras que en el caso de *INTERTEG*, como ya se vio anteriormente, no siempre los resultados son adecuados.

En resumen, se puede afirmar que con el código *INTPOLBI* se obtienen buenos resultados para el parámetro k infinita, mientras que en el caso de *INTERTEG*, se necesitan ciertas condiciones para que esto suceda. Adicionalmente, aún en los casos en los cuales los resultados de *INTERTEG* son buenos, los errores relativos obtenidos con *INTPOLBI* son menores, esto es, los valores de k infinita se acercan con mayor exactitud al emplear dicho código.

Con respecto a los resultados del parámetro k infinita, comparando las dos versiones de *INTPOLBI*, se puede observar de las Tablas 6.5 a 6.14, que no existe gran diferencia, de hecho, los resultados prácticamente son iguales y en algunos casos la primera versión supera (en realidad por muy poco) a la segunda, este hecho podría entenderse como una contradicción con respecto a lo que se ha venido afirmando con respecto a la mejoría de los resultados al emplear la segunda versión, sin embargo, no es tal debido a que la mejoría de resultados se observa claramente en los valores obtenidos para el total de parámetros, lo cual se analizará a continuación.

Antes de hacer alguna conclusión, es conveniente mencionar que, al revisar los resultados globales de la interpolación con ambos códigos y sus respectivas variantes, se pudo observar lo siguiente, si bien en las tablas respectivas (6.15 y 6.16) solo se incluyen uno o dos "casos" de los seis que en realidad son al obtener un nuevo Banco, los resultados ahí mostrados son similares en todos los casos, por otra parte, y como ya se ha reiterado, el resumen mostrado en las tablas mencionadas refleja fielmente los resultados de todas las pruebas realizadas.

Una vez mencionado lo anterior se pueden hacer algunas afirmaciones, ya se dijo que para los usuarios de los Bancos, bastaba con obtener errores relativos del orden de 10^{-3} , para considerar que el Banco obtenido satisfacía las necesidades básicas y por ende poder trabajar con dichos resultados, por tal motivo, se consideró necesario poner una cota más pequeña, cuyo valor es de 10^{-5} y la cual se emplea para verificar los resultados globales⁸, en ese sentido, se puede verificar que para el código *INTPOLBI* en promedio se tiene que entre el 30 y 40 por ciento de los valores interpolados se encuentran por debajo de la cota convenida, lo cual nos permite afirmar que los resultados son adecuados, adicionalmente, y aunque no se refleja en las tablas mencionadas, es importante mencionar de los once parámetros más importantes del Banco, la mitad cae en la situación anterior⁹, y los restantes tienen errores relativos entre 10^{-3} y 10^{-4} , lo cual se puede considerar como una muy buena aproximación.

En cuanto a *INTERTEG*, se pueden repetir de manera automática los comentarios hechos al analizar los resultados de k infinita, esto es, se vuelve indispensable que se cumplan ciertas condiciones para que los resultados sean buenos, pues si se toma como referencia el caso en el cual se tenían 8 Bancos de entrada (Tabla 6.16, Bancos 8 y 9), se puede verificar que no se tiene un solo error relativo menor a la cota propuesta, de hecho, al verificar los valores interpolados, se tiene que

⁸ De los 840 valores que se interpolan, existen alrededor de 60 parámetros cuyo valor es cero, los cuales no se incluyen en las tablas mencionadas, debido a que, por obvias razones los comentarios hechos tendrían cierta parcialidad.

⁹ Hubo ejemplos en los cuales la totalidad de los parámetros mencionados estaban por debajo de la cota referida.

aproximadamente el 95 por ciento de los errores relativos que se calculan para dichos parámetros, tienen valores entre 10^0 y 10^1 , lo cual a todas luces es un mal resultado.

En este punto es en el cual se puede ver la diferencia entre las dos versiones de *INTPOLBI*, pues de las Tablas 6.15 y 6.16 se puede verificar que si bien en algunos ejemplos la tendencia es similar en ambas versiones, en un poco menos de la mitad de éstos, aumenta el número de casos en los cuales el error relativo es menor a la cota propuesta; aún más, de las pruebas realizadas, se observó que en 2 de cada 10 Bancos obtenidos, el aumento en el número de ocasiones en los cuales el error relativo era menor a la cota mencionada era de aproximadamente el doble entre las dos versiones de *INTPOLBI*, lo cual es una mejora considerable.

También en las tablas mencionadas (6.15 y 6.16), se incluyen los errores relativos máximo y mínimo, del total de datos de un Banco, incluyendo todos sus "casos", de dichos valores se puede simplemente comentar los siguiente, en todos los casos, el error relativo máximo se encontró en un parámetro de poca importancia e influencia dentro del Banco, que no obstante, aún con éste comentario, se puede observar que dichos valores se encuentran en límites adecuados al emplear cualquiera de las dos versiones de *INTPOLBI*. Nuevamente, y en la misma situación, con *INTERTEG* se tienen resultados erróneos en los Bancos 8 y 9 (Tabla 6.16).

Adicionalmente a los comentarios y conclusiones anteriores, siempre que se concluye un sistema de cómputo, conviene saber qué tan "caro" resulta el mismo, al calcular el tiempo de máquina invertido para llevar a cabo el proceso, en este caso, al observar la Tabla 6.17, se puede observar que al emplear el código *INTERTEG*, el tiempo de CPU es menor con respecto al mismo dato en *INTPOLBI*, independientemente del número de Bancos de entrada empleados, lo cual en primera instancia podría verse como una ventaja importante, sin embargo, conviene señalar algunas cuestiones importantes para tener bases más sólidas, al afirmar o desechar tal afirmación. Una diferencia importante entre ambos códigos es la siguiente, el código *INTPOLBI* realiza una serie de validaciones antes y después del proceso de interpolación, como son, la ordenación de los Bancos, algunos datos para verificar que los Bancos sean del mismo tipo, el condicional de la matriz obtenida (para evitar resultados erróneos en el Banco de salida), entre otras, mientras que el código *INTERTEG* no realiza ninguna validación de los datos de entrada, lo cual incrementa las posibilidades de error. Por otra parte, el código *INTPOLBI* calcula el condicional de la matriz, los residuales de todos los parámetros del Banco y almacena los coeficientes de todos los polinomios

encontrados en el proceso al momento de interpolar¹⁰, lo cual produce que se incremente de manera considerable los cálculos realizados; nuevamente, el código *INTERTEG* no realiza ningún otro cálculo, fuera de la interpolación de Lagrange.

Adicionalmente, es necesario decir que para llevar a cabo la interpolación con *INTERTEG*, basta con tener dos Bancos del mismo tipo y el proceso es inmediato, mientras que para *INTPOLBI* se necesitan 16 Bancos de entrada para generar la malla que se requiere, aún más, deben estar distribuidos adecuadamente, para evitar problemas, todo esto puede ser una desventaja adicional para el código *INTPOLBI*, sin embargo, veamos lo siguiente, una vez que se tienen los 16 Bancos de entrada para *INTPOLBI*, si se asegura que se encuentran distribuidos uniformemente dentro de los valores que puede haber tanto para uranio, como para gadolinia, entonces se puede obtener cualquier Banco sin necesidad de modificar la malla, aún más, con los coeficientes que se almacenan en un archivo en la primera interpolación, se pueden generar los Bancos que se quieran empleando el módulo *EVALUA* del sistema y con resultados igual de confiables, con la ventaja de que el tiempo de CPU se compara con el empleado para *INTERTEG*, éste último además, tiene la desventaja de que si se desea otro Banco en un intervalo diferente, también se necesitarán nuevos Bancos de entrada, adicionalmente, y como ya se demostró, es necesario que los Bancos de entrada estén lo suficientemente cercanos (en cuanto al porcentaje del parámetro de interpolación), para obtener buenos resultados, en caso contrario, no se puede asegurar mucho; en resumen, se puede concluir que la aparente desventaja en el tiempo de CPU invertido con *INTPOLBI*, no es tal, si se toma en cuenta la versatilidad que tiene al conjuntar el módulo *EVALUA* y la facilidad con la que se puede obtener un nuevo Banco sin necesidad de modificar los datos de entrada.

Es necesario mencionar que, si bien el código *INTERTEG* tiene ciertos problemas en algunos casos como se demostró anteriormente, también se tiene que, cuando se cumplen ciertas condiciones, los resultados obtenidos son buenos como se puede observar en las Figuras 6.3 y 6.7, de hecho, en general y de acuerdo a las pruebas que se estuvieron realizando, si se cumple que los valores del parámetro de interpolación son cercanos y no existen cambios bruscos en los valores de los parámetros de los Bancos de entrada, los resultados son adecuados. En conclusión, existen muchos casos en los cuales se cumplen las condiciones anteriores y por lo tanto en dichos casos el código funciona bien.

Veamos ahora, en la Tabla 6.18 se muestran los resultados obtenidos para el número de condición de la matriz obtenida en algunos ejemplos trabajados, durante las pruebas de validación de las dos versiones de *INTPOLBI*, claramente se

¹⁰ Existen tantos polinomios como datos en el Banco, pues para cada parámetro el polinomio de interpolación cambia.

tiene que la diferencia entre ambos es considerable, de donde se deben comentar dos aspectos importantes, primero: ya se mencionó que la matriz obtenida en la segunda versión es menos sensible que la primera, por las bases empleadas al momento de obtener el polinomio interpolante, lo cual produce los resultados mencionados; segundo: de los comentarios y conclusiones hechas anteriormente, se observa el beneficio que se obtiene al emplear la segunda versión.

Finalmente, es importante señalar que el código es perfectible aún en muchos aspectos, de hecho, se puede trabajar en mejoras a la interpolación o en métodos completamente alternos. Sin embargo, la propuesta aquí presentada puede servir de base en cualquiera de los caminos anteriores, pues hasta el momento, el código ha demostrado ser una buena alternativa para los fines que se persiguieron desde que fue concebido.

Apéndice A

Manual de Usuario

Para dar por concluido el presente trabajo, a continuación se incluye el manual de usuario elaborado para entender el funcionamiento del sistema *INTPOLBI*, en este apéndice no se incluye ningún concepto técnico importante, simplemente se explican los pasos a seguir para la ejecución del sistema mencionado, asimismo se explican las posibles salidas en caso de alguna falla, ya sea al momento de introducir algún parámetro de entrada, o por algún error del sistema de cómputo empleado¹.

En el presente manual se indican los requisitos necesarios para poder ejecutar de manera óptima el sistema *INTPOLBI*, así como las restricciones que se tienen y los pasos a seguir en caso de se presente alguna falla. En cada uno de los pasos se indican los parámetros de entrada requeridos por el sistema, con las limitantes respectivas.

¹ En realidad se refiere a una posible interrupción de la energía eléctrica, por alguna falla de la red de cómputo o cuestiones de este tipo.

Para una mejor comprensión, el presente manual se divide en dos partes: la primera de ellas se refiere a la obtención de un Banco de Datos Nucleares a partir de 16 ya existentes utilizando el método de interpolación mencionado a lo largo del presente trabajo; en la segunda parte se explica la forma de generar un nuevo Banco con la información que se obtuvo de una interpolación previa. Esto se debe a que el sistema guarda los coeficientes de los polinomios obtenidos en cada interpolación, para después simplemente evaluar en cada uno de estos. A su vez, cada una de las partes mencionadas se subdivide en cuatro partes: lectura, ejecución, resultados y errores. En algunas de ellas la explicación se profundiza de manera especial debido a la importancia que tienen los conceptos que ahí se tratan.

El sistema se implementó finalmente en el equipo MicroVax 3100-80 de la Gerencia de Ciencias Aplicadas del Instituto Nacional de Investigaciones Nucleares (ININ), para uso de quién así lo solicite.

A.1.- Requisitos

Para poder emplear el sistema *INTPOLBI*, es necesario tener cubiertos tres requisitos fundamentales, primero: tener acceso al equipo mencionado; en segundo término, contar con al menos 16 Bancos del mismo tipo y de los cuales se conozcan sus porcentajes uranio y gadolinia; por último, crear un archivo ASCII que contenga los parámetros de entrada, el contenido de este último se explicará mas adelante.

A continuación se explicará el funcionamiento de cada uno de los módulos del sistema, para ello, es necesario aclarar que todas las Figuras incluidas en este apéndice se refieren a los enunciados que aparecen en pantalla al momento de ejecutar el sistema, esto puede variar dependiendo de la terminal en la cual se trabaje. El despliegue mostrado aquí, esta basado en un ejemplo real tomando como base un conjunto de 16 Bancos de entrada.

A.2.- Módulo de Interpolación

Por ésta única vez se mencionará que es necesario abrir una sesión de trabajo en el equipo MicroVax arriba mencionado, para poder ejecutar cualquiera de los módulos del sistema *INTPOLBI*. Una vez abierta la sesión de trabajo

correspondiente, la ejecución del módulo de interpolación se hace tecleando la siguiente instrucción²:

\$ @ INTERPOLA ↵

Una vez tecleada la instrucción anterior, aparece el primer mensaje del sistema, que consiste de un "banner", indicando el nombre del sistema y la versión, acompañado de un mensaje de pausa que detiene la ejecución del sistema hasta que el usuario oprima la tecla de ENTER para continuar. La Figura A.1 muestra el mensaje mencionado anteriormente.

```

          INSTITUTO NACIONAL DE INVESTIGACIONES NUCLEARES

IIII NNN  NN TTTTTT EEEE  RRRR  PPPP  OOO  LL  AAA
II  NNNN NN  TT  EE  RR RR  PP PP  OO OO LL  AA AA
II  NNNNNNN  TT  EEE  RRRR  PPPP  OO OO LL  AAAAA
II  NN NNNN  TT  EE  RR RR  PP  OO OO LL  AA AA
IIII NN  NNN  TT  EEEE  RR RR  PP  OOO  LLLL AA AA

          =====
          =====

          VV      VV          11          000
            VV    VV          111         00 00
              VV  VV          11          00 00
                VVV          11  ..     00 00
                  V          1111  ..     000

Pausa .... Oprime RETURN para continuar
    
```

Figura A.1. "Banner" del módulo de interpolación.

² El símbolo empleado al termino de la instrucción, indica que se debe oprimir la tecla de ENTER, esta nomenclatura se utilizará en todo el apéndice.

Al oprimir la tecla de **ENTER** en la pausa anterior, se inicia la lectura de los parámetros de entrada. El primero parámetro que solicita el sistema es el nombre del archivo de entrada, para lo cual se despliega el mensaje mostrado en la Figura A.2. El archivo solicitado debe cumplir los siguientes requisitos:

- a) El nombre debe tener a lo más 20 caracteres.
- b) Que el archivo se encuentre en el directorio de trabajo³.

Si se cumplen los requisitos anteriores el programa continúa normalmente (el sistema verifica no sólo la existencia del archivo, sino también su contenido), en caso contrario pueden ocurrir cuatro posibilidades, a continuación se indican estas, es necesario señalar que en cualesquiera de las situaciones anteriores, la ejecución del sistema se detiene automáticamente.

```
PROPORCIONA EL NOMBRE DEL ARCHIVO DE ENTRADA
=====
recuerda que dicho archivo contiene en la
primera línea el número de casos de los
Bancos a interpolar, en las siguientes los
identificadores de las celdas y por último
los nombres de los archivos acompañados de
su respectivo porcentaje de gadolinia:
```

Figura A.2. Captura del archivo de entrada.

- a) Si el archivo proporcionado no existe ya sea en el directorio de trabajo o en la trayectoria indicada, se despliega un mensaje de error, la Figura A.3 muestra dicho mensaje.

```
----> ERROR 1 EN LA APERTURA DEL ARCHIVO DE ENTRADA
```

Figura A.3. Mensaje de error en la apertura del archivo de entrada.

- b) Si el archivo proporcionado existe pero no tiene la información requerida, se despliega el mensaje de error mostrado en la Figura A.4.

³ Es válido proporcionar trayectorias, siempre y cuando no excedan el número de caracteres indicado.

----> ERROR 2 EN LA LECTURA DEL ARCHIVO DE ENTRADA

Figura A.4. Mensaje de error en la lectura del archivo de entrada.

- c) Si el archivo proporcionado contiene menos de 16 nombres de Bancos de entrada, se despliega el mensaje de error mostrado en la Figura A.5.

----> ERROR 3 FALTAN ARCHIVOS DE BANCOS (TIENEN QUE SER 16)

Figura A.5. Mensaje de error en el número de Bancos proporcionado.

- d) Si el archivo proporcionado tiene más de 16 nombres de Bancos se despliega el mensaje de error mostrado en la Figura A.6.

----> ERROR 4 SOBRAN ARCHIVOS DE BANCOS (TIENEN QUE SER 16)

Figura A.6. Mensaje de error en el número de Bancos proporcionado.

El segundo parámetro solicitado es el nombre del Banco de salida, para lo cual se despliega el mensaje mostrado en la Figura A.7; al igual que el archivo anterior solamente se tomarán en cuenta los primeros 20 caracteres para el nombre, los restantes (si es el caso) se eliminan. Si por alguna circunstancia no se proporciona algún nombre, el sistema le asigna *FOR003.DAT*.

DAME EL NOMBRE DEL BANCO DE SALIDA:

Figura A.7. Lectura del nombre del Banco de salida.

El siguiente parámetro que solicita el sistema, es el nombre del archivo donde se guardarán los coeficientes obtenidos durante la interpolación, en este caso, se despliega el mensaje mostrado en la Figura A.8. Análogamente a los casos

anteriores, sólo se tomarán en cuenta los primeros 20 caracteres para el nombre, los restantes (si es el caso) se eliminan. Si por alguna circunstancia no se proporciona nombre alguno, el sistema le asignará el de *FOR004.DAT*.

**DAME EL NOMBRE DEL ARCHIVO DONDE SE
GUARDARAN LOS COEFICIENTES DEL
POLINOMIO RESULTANTE:**

Figura A.8. Lectura del nombre del archivo de coeficientes.

Antes de continuar con la explicación, conviene hacer notar que es necesario tener cuidado al manejar este archivo, ya que posteriormente se puede utilizar en el segundo módulo del sistema para la obtención de nuevos Bancos, por tal motivo se recomienda asignarle un nombre de acuerdo al conjunto de Bancos de entrada utilizado, con el objeto de no olvidarlo posteriormente.

A.2.1.- Archivo de Entrada

De los parámetros solicitados en esta primera parte, el más importantes es el archivo de entrada, debido a que es el medio utilizado para proporcionarle al sistema los tres principales parámetros para realizar la interpolación, a saber, el número de "*casos*" de los Bancos involucrados, los identificadores para cada "*caso*" y por último, los nombres de los Bancos utilizados para la interpolación. Por todo lo anterior, es necesario explicar detalladamente la estructura de dicho archivo, a continuación se describe ésta incluyendo el formato en el cual debe ir la información en el archivo mencionado.

En la primera línea se escribe el valor entero asociado al número de "*casos*"⁴ del Banco de salida, esta cantidad debe coincidir con el número de "*casos*" de los Bancos de entrada. El formato para escribir este valor es libre, siempre y cuando sea entero.

En las siguientes 18 líneas se escriben los identificadores para cada una de las celdas del Banco de salida, colocando dos identificadores por línea, tomando en

⁴ En el presente apéndice no se abundará acerca de los "*casos*" de un Banco, debido a que ya se ha mencionado en capítulos anteriores.

cuenta que cada celda ocupa dos de éstos. El formato empleado debe ser de 10 caracteres para cada identificador, iniciando en la primera columna y sin dejar espacios entre ellos, esto es, en total se tendrán 20 caracteres en cada línea.

Por último, en las 16 líneas restantes se colocan los nombres de los Bancos de entrada⁵, acompañados de su respectivo porcentaje de gadolinia. Los nombres de los Bancos deben iniciar en la primera columna y utilizar a lo más 12 caracteres⁶. El valor para la gadolinia se coloca entre las columnas 13 y 20 en formato libre; de hecho, aún cuando el valor de la gadolinia es real, se puede colocar un valor entero sin problema alguno, el sistema realiza la transformación correspondiente de manera interna.

En la Figura A.9 se muestra un archivo de entrada típico; para una mejor comprensión por parte del lector, se hace la observación de que las cursivas en dicha Figura son para indicar lo anteriormente expuesto, y en ningún momento forman parte del archivo. Finalmente, es necesario remarcar que el formato arriba mencionado es de suma importancia para el buen funcionamiento del sistema, debido a que, como ya se mencionó anteriormente, se verifica tanto la existencia del archivo, como su contenido.

Con lo anterior, se da por concluida la discusión acerca de la estructura interna del archivo de entrada, en los párrafos siguientes se continúa con la explicación de los parámetros de entrada solicitados por el sistema.

Una vez que se proporcionan los nombres de los archivos requeridos, el sistema continúa con la lectura de los parámetros de entrada, el siguiente de estos es el porcentaje de gadolinia para el Banco de salida. La Figura A.10 muestra el mensaje que se despliega en la pantalla, para llevar a cabo esto. Cabe hacer notar, que aún cuando el parámetro mencionado es de tipo real, se puede proporcionar un valor entero sin ningún problema, el sistema se encargará de realizar la transformación correspondiente.

⁵ Aún cuando pudiera parecer reiterativo, conviene señalar que estos Bancos deben ser del mismo tipo.

⁶ Este número de caracteres ya incluye la extensión.

18				<i># de casos</i>
	RI	SM	041 00	<i>identificadores</i>
	RI	SM	041 01	.
	RI	SM	041 02	.
	RI	SM	042 00	.
	RI	SM	042 01	.
	RI	SM	042 02	.
	RI	SM	043 00	.
	RI	SM	043 01	.
	RI	SM	044 00	.
	RI	SM	045 00	.
	RI	SM	045 01	.
	RI	SM	045 02	.
	RI	SM	045 03	.
	RI	SM	045 04	.
	RI	SM	045 05	.
	RI	SM	045 06	<i>identificadores</i>
	L1REM101.REC		21	<i>Banco de entrada</i>
	L1REM102.REC		7	.
	L1REM103.REC		28	.
	L1REM104.REC		14	.
	L1REM201.REC		21	.
	L1REM202.REC		7	.
	L1REM203.REC		28	.
	L1REM204.REC		14	.
	L1REM301.REC		21	.
	L1REM302.REC		7	.
	L1REM303.REC		28	.
	L1REM304.REC		14	.
	L1REM401.REC		21	.
	L1REM402.REC		7	.
	L1REM403.REC		28	.
	L1REM404.REC		14	<i>Banco de entrada</i>

Figura A.9. Archivo de entrada.

% DE GADOLINIA PARA EL BANCO DE SALIDA:

Figura A.10. Lectura del porcentaje de gadolinia.

El parámetro anterior debe estar en el rango permitido, esto es, entre 0.8 y 50.0, en caso contrario, el sistema despliega el mensaje de error mostrado en la Figura A.11, para posteriormente terminar la ejecución.

--> ERROR 12 EL PORCENTAJE DE GADOLINIA ESTA FUERA DEL RANGO PERMITIDO (0.8 < Gd₂O₃ < 50.0)

Figura A.11. Error en el porcentaje de gadolinia.

Al concluir la lectura anterior, el sistema despliega el mensaje mostrado en la Figura A.12, solicitando con ello el porcentaje de uranio para el Banco de salida; al igual que el parámetro anterior, el porcentaje de uranio solicitado es una variable de tipo real, no obstante, el usuario puede proporcionar un valor con formato entero y el sistema se encargará de realizar la transformación correspondiente.

% DE ²³⁵U PARA EL BANCO DE SALIDA:

Figura A.12. Lectura del porcentaje de uranio.

El valor proporcionado por el usuario debe estar en el rango permitido, esto es, entre 0.7 y 4.2, de lo contrario, el sistema despliega un mensaje de error (Figura A.13) terminando con ello la ejecución.

--> ERROR 13 EL PORCENTAJE DE URANIO ESTA FUERA DEL RANGO PERMITIDO (0.7 < ²³⁵U < 4.2)

Figura A.13. Error en el porcentaje de uranio.

Al concluir la lectura de los datos de entrada, el sistema despliega en pantalla los datos leídos (Figuras A.14-A.17) para verificar si son los adecuados para llevar a cabo la interpolación. El despliegue de los datos de entrada se divide en cuatro partes, primero el letrero que indica el inicio de la presentación; en segundo plano los nombres de los Bancos de entrada acompañados de su respectivo porcentaje de gadolinia y uranio; en tercer término los identificadores para cada uno de los "casos" del Banco de salida y por último, se despliegan los porcentajes de gadolinia y uranio del Banco de salida, así como el número de "casos" incluidos en éste. Cada una de las partes mencionadas va acompañada de la pausa correspondiente. El orden de las Figuras es el mismo en el que aparecen durante el proceso.

LOS PARAMETROS DE ENTRADA SON LOS SIGUIENTES:
Pausa ... oprime RETURN para continuar

Figura A.14. Mensaje inicial para desplegar datos de entrada.

```

=====
ARCHIVOS DE BANCOS, % RESPECTIVO DE Gd2O3 Y % de 235U

L1REM101.REC      21.0      3.8093
L1REM102.REC       7.0      3.8093
L1REM103.REC      28.0      3.8093
L1REM104.REC      14.0      3.8093
L1REM201.REC      21.0      3.6092
L1REM202.REC       7.0      3.6092
L1REM203.REC      28.0      3.6092
L1REM204.REC      14.0      3.6092
L1REM301.REC      21.0      3.7092
L1REM302.REC       7.0      3.7092
L1REM303.REC      28.0      3.7092
L1REM304.REC      14.0      3.7092
L1REM401.REC      21.0      3.9092
L1REM402.REC       7.0      3.9092
L1REM403.REC      28.0      3.9092
L1REM404.REC      14.0      3.9092
=====

Pausa ... oprime RETURN para continuar
    
```

Figura A.15. Despliegue de los Bancos de entrada.

```

=====
IDENTIFICADORES PARA LAS CELDAS DEL NUEVO BANCO

      RI   SM   <->   041 00
      RI   SM   <->   041 01
      RI   SM   <->   041 02
      RI   SM   <->   042 00
      RI   SM   <->   042 01
      RI   SM   <->   042 02
      RI   SM   <->   043 00
      RI   SM   <->   043 01
      RI   SM   <->   044 00
      RI   SM   <->   045 00
      RI   SM   <->   045 01
      RI   SM   <->   045 02
      RI   SM   <->   045 03
      RI   SM   <->   045 04
      RI   SM   <->   045 05
      RI   SM   <->   045 06
=====

Pausa ... oprime RETURN para continuar
    
```

Figura A.16. Despliegue de los identificadores de las celdas.

```

PARAMETROS DEL NUEVO BANCO

      GADOLINIA = 18.45
      URANIO    = 3.8945
      # DE CASOS = 18

=====

ESTAN CORRECTOS LOS PARAMETROS DE ENTRADA (S/N)?
    
```

Figura A.17. Despliegue de los parámetros de entrada.

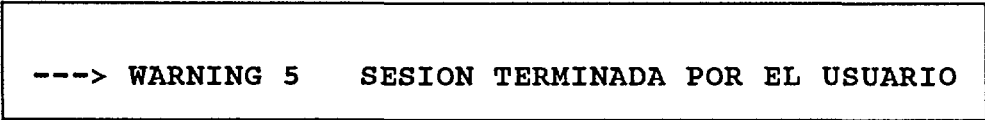
Al desplegar la última parte de la información de entrada, el sistema solicita una respuesta afirmativa (tecleando la letra "S" en mayúsculas o minúsculas) en caso de que la información desplegada sea la correcta para continuar con el proceso; si es el caso, se despliega el mensaje mostrado en la Figura A.18, indicando que se está procesando la información, es decir, en este momento se inicia el proceso de interpolación.



Un momento estoy procesando

Figura A.18. Información durante el proceso.

Si se considera que la información no es la correcta, esto es, se responde negativamente a la pregunta anterior (con una letra "N" en mayúsculas, minúsculas o cualquier otro carácter distinto de "S"), el sistema concluye la ejecución del mismo, desplegando el mensaje mostrado en la Figura A.19. Para modificar alguno de los parámetros proporcionados es necesario volver a ejecutar desde el principio el sistema.



---> WARNING 5 SESION TERMINADA POR EL USUARIO

Figura A.19. Mensaje si la respuesta es negativa.

Si el sistema no detecta ningún error durante la ejecución, termina desplegando los nombres de los archivos obtenidos durante el proceso de interpolación, el mensaje que aparece en la pantalla se muestra en la Figura A.20. La información contenida en cada uno de los archivos de salida se describe a continuación.

BANCO DE SALIDA : Es un archivo binario que contiene el Banco de Datos Nucleares obtenido mediante el proceso de interpolación, a partir de los 16 Bancos de entrada proporcionados.

ARCHIVO DE COEFICIENTES : Es un archivo binario que contiene los coeficientes de cada uno de los polinomios interpolantes obtenido durante el proceso. Es importante resaltar que se tiene un polinomio por cada parámetro interpolado

```

*****
*
*      EL PROGRAMA DE INTERPOLACION      *
*      FINALIZO SIN PROBLEMAS            *
*      !!!! VERIFIQUE SUS RESULTADOS !!!! *
*
*      BANCO DE SALIDA : L1REM999.REC    *
*
*      ARCHIVO CON LOS COEFICIENTES DE LA *
*      INTERPOLACIÓN:      L1REM999.COEF *
*
*****
    
```

Figura A.20. Mensaje para indicar el fin de la ejecución.

A.2.2.- Errores del Módulo

Durante el proceso de ejecución pueden ocurrir errores que provoquen que ésta termine, con lo cual los resultados no son satisfactorios. En esta sección se hace un resumen de aquellos errores que pueden ocurrir durante la ejecución y que no han sido mencionados hasta el momento (Figuras A.21-A.26), acompañado en cada caso del mensaje que despliega el sistema, más una breve explicación de la causa y finalmente la posible solución.

```

--> ERROR 6  EL # DE CASOS DE LOS BANCOS DE ENTRADA NO
              COINCIDE CON EL DEL NUEVO BANCO
    
```

Figura A.21. Mensaje de error.

El sistema detectó que el número de "casos" del Banco de salida, no coincide con el número de "casos" de alguno(s) de los Bancos de entrada. Es necesario editar el archivo de entrada y corregir el dato correspondiente.

```

--> ERROR 7  EN LA APERTURA DEL BANCO DE ENTRADA
    
```

Figura A.22. Mensaje de error.

Alguno de los Bancos de entrada proporcionados no se encuentra en el directorio de trabajo (o en el directorio especificado) o el archivo proporcionado no es realmente un Banco. Verifique la existencia de los Bancos de entrada, así como el formato.

--> ERROR 8 EN LA LECTURA DEL BANCO DE ENTRADA

Figura A.23. Mensaje de error.

Alguno de los archivos proporcionados como Bancos de entrada contiene información distinta de la que realmente tiene un Banco. Verifique que los archivos proporcionados realmente son Bancos de Datos Nucleares.

--> ERROR 9 EL BANCO DE SALIDA ES IGUAL (DE ACUERDO A LOS % DE ^{235}U Y Gd_2O_3) A UNO DE LOS BANCOS DE ENTRADA

Figura A.24. Mensaje de error.

Se proporcionaron porcentajes de uranio y gadolinia iguales a los que contiene uno de los Bancos de entrada. Verifique los porcentajes de uranio y gadolinia de los Bancos de entrada y ejecute nuevamente el sistema proporcionando los valores adecuados.

--> ERROR 10 LA MATRIZ DE ^{235}U Y Gd_2O_3 ES MAL CONDICIONADA POR LO CUAL EL BANCO DE SALIDA NO ESTA COMPLETO

Figura A.25. Mensaje de error.

Los Bancos de entrada proporcionados, no están lo suficientemente dispersos (refiriéndose a la "malla" que se forma graficando los porcentajes de uranio y gadolinia de los 16 Bancos). Idealmente se propone que la malla formada por los valores mencionados sea una matriz de 4x4 exactamente. En este caso es necesario

cambiar algunos de los Bancos de entrada de tal forma que se modifique la estructura de la matriz.

```
--> ERROR 11  EN LA APERTURA DEL ARCHIVO DE SALIDA  
              DONDE SE GUARDAN LOS COEFICIENTES DE  
              INTERPOLACION
```

Figura A.26. Mensaje de error.

El archivo de salida donde se guardarán los coeficientes de los polinomios de interpolación no se pudo abrir. Generalmente ocurre cuando el equipo tiene algún problema de memoria o espacio en disco, lo cual es muy poco común. En este caso basta con él restablecimiento del equipo para que el sistema funcione normalmente.

A.3.- Módulo de Evaluación

Como se mencionó al inicio, resulta más sencillo utilizar la información obtenida en una interpolación previa, si se desea obtener un nuevo Banco, bajo la premisa de que será del mismo tipo. Esto se lleva a cabo ejecutando el módulo de evaluación del sistema *INTPOLBI*, empleando los porcentajes de uranio y gadolinia del nuevo Banco. Para tener consistencia en la explicación, se utilizó el mismo ejemplo del módulo anterior. Veamos ahora, para ejecutar el módulo de evaluación es necesario teclear la siguiente instrucción:

```
$ @ EVALUA_POLI ↵
```

Una vez tecleada la instrucción anterior, el sistema despliega en la pantalla el "banner" de presentación mostrado en la Figura A.27, indicando el nombre del mismo y la versión respectiva; al final del mensaje aparece una pausa que detiene la ejecución hasta que se oprima la tecla de ENTER.

Al oprimir la tecla de ENTER en la pausa anterior, el sistema inicia la lectura de los parámetros de entrada; el primer dato solicitado es el nombre del archivo de entrada, para lo cual se despliega el mensaje mostrado en la Figura A.28. Se omite en este caso la explicación de la lectura debido a que es la misma que la presentada anteriormente, incluyendo los posibles errores y los mensajes respectivos.

```

                INSTITUTO NACIONAL DE INVESTIGACIONES NUCLEARES

EEEE      VV   VV      AAA      LL      UU UU      AAA
EE        VV   VV      AA AA     LL      UU UU      AA AA
EEE       VV   VV      AAAAA     LL      UU UU      AAAAA
EE        VV   VV      AA AA     LL      UU UU      AA AA
EEEE      VVV        AA AA     LLLL     UUU      AA AA

                =====
                =====

                VV      VV      11      000
                VV   VV      111     00 00
                VV   VV      11      00 00
                VVV        11      .. 00 00
                V          1111   .. 000

Pausa .... Oprime RETURN para continuar
    
```

Figura A.27. "Banner" del módulo de evaluación.

```

PROPORCIONA EL NOMBRE DEL ARCHIVO DE ENTRADA
=====
                !!!! I M P O R T A N T E !!!!
=====
el archivo debe ser el mismo que el utilizado
en el proceso de interpolación:
    
```

Figura A.28. Captura del archivo de entrada.

El segundo dato solicitado por el sistema, es el nombre del archivo donde se guardaron los coeficientes obtenidos en una interpolación previa; en este caso, se despliega el mensaje mostrado en la Figura A.29. En forma análoga al anterior módulo, solo se tomarán en cuenta los primeros 20 caracteres para el nombre del archivo en cuestión, los restantes, si aparecen, no se toman en cuenta.

**DAME EL NOMBRE DEL ARCHIVO DE
COEFICIENTES:**

Figura A.29. Lectura del archivo de coeficientes.

Si el archivo proporcionado fue el correcto, el sistema continúa la ejecución normalmente, en caso contrario, el sistema despliega el mensaje de error mostrado en la Figura A.30, terminando con ello la ejecución.

**--> ERROR 11 EN LA APERTURA DEL ARCHIVO DE COEFICIENTES
DE INTERPOLACION**

Figura A.30. Mensaje de error en la apertura del archivo de coeficientes.

El dato solicitado, una vez proporcionado el archivo de coeficientes, es el nombre de uno de los Bancos utilizados en la interpolación previa, esto es, cualquiera de los 16 Bancos que se emplearon en la interpolación original; para llevar a cabo esto, el sistema despliega el mensaje mostrado en la figura A.31.

**DAME EL NOMBRE DE UNO DE LOS BANCOS
UTILIZADOS EN LA INTERPOLACION:**

Figura A.31. Lectura del Banco de entrada.

Si el Banco proporcionado es el adecuado, el sistema continúa el proceso de evaluación, en caso contrario, pueden ocurrir cualesquiera de las dos posibilidades siguientes: primero, si el Banco proporcionado no se encuentra en el directorio de trabajo, se despliega el mensaje de error mostrado en la Figura A.32; la segunda

posibilidad es que el Banco proporcionado si se encuentre en el directorio mencionado, pero que no sea uno de los utilizados en el proceso de interpolación previa⁷, en cuyo caso, se despliega el mensaje de error mostrado en la Figura A.33. En cualquiera de las situaciones anteriores, se termina la ejecución del sistema.

--> ERROR 7 EN LA APERTURA DEL BANCO DE ENTRADA

Figura A.32. Mensaje de error en la apertura del Banco de entrada.

--> ERROR 10 EL BANCO DE ENTRADA PROPORCIONADO NO ES UNO DE LOS UTILIZADOS EN LA INTERPOLACION

Figura A.33. Mensaje de error si el Banco de entrada no es el adecuado.

En lo referente a la lectura de archivos, el último dato solicitado es el nombre del Banco de salida, para lo cual se despliega el mensaje mostrado en la Figura A.34; en este caso solamente se tomarán en cuenta los primeros 20 caracteres para el nombre, los restantes, si se proporcionan, no se tomarán en cuenta. Si por alguna circunstancia, no se proporciona algún nombre, el sistema le asigna el de FOR003.DAT.

DAME EL NOMBRE DEL BANCO DE SALIDA:

Figura A.34. Lectura del nombre del Banco de salida.

No se abundará mucho en esta sección con respecto a los mensajes de error, debido a que, son los mismos que aparecen en el módulo anterior.

Al concluir la lectura de los parámetros de entrada, el sistema despliega en pantalla los datos leídos, apareciendo para ello los mensajes mostrados en las Figuras de la A.35 a la A.38, con el objeto de verificar si son los adecuados para llevar a cabo la evaluación correspondiente.

⁷ De hecho, el sistema lo verifica de acuerdo a los nombres incluidos en el archivo de entrada.

```
LOS PARAMETROS DE ENTRADA SON LOS SIGUIENTES:
```

```
  Pausa ... oprime RETURN para continuar
```

Figura A.35. Mensaje inicial para desplegar datos de entrada.

```
=====
LOS COEFICIENTES DE INTERPOLACIÓN SE OBTUVIERON CON
LOS SIGUIENTES BANCOS:
```

```
L1REM101.REC
L1REM102.REC
L1REM103.REC
L1REM104.REC
L1REM201.REC
L1REM202.REC
L1REM203.REC
L1REM204.REC
L1REM301.REC
L1REM302.REC
L1REM303.REC
L1REM304.REC
L1REM401.REC
L1REM402.REC
L1REM403.REC
L1REM404.REC
```

```
=====
Pausa ... oprime RETURN para continuar
```

Figura A.36. Despliegue de los Bancos utilizados en la interpolación.

```

=====
IDENTIFICADORES PARA LAS CELDAS DEL NUEVO BANCO

      RI   SM  <->   041 00
      RI   SM  <->   041 01
      RI   SM  <->   041 02
      RI   SM  <->   042 00
      RI   SM  <->   042 01
      RI   SM  <->   042 02
      RI   SM  <->   043 00
      RI   SM  <->   043 01
      RI   SM  <->   044 00
      RI   SM  <->   045 00
      RI   SM  <->   045 01
      RI   SM  <->   045 02
      RI   SM  <->   045 03
      RI   SM  <->   045 04
      RI   SM  <->   045 05
      RI   SM  <->   045 06
=====

Pausa ... oprime RETURN para continuar
    
```

Figura A.37. Despliegue de los identificadores de las celdas.

```

          BANCO DE ENTRADA: L1REM101.REC
% DE Gd2O3 DEL BANCO DE ENTRADA: 23.000
% DE 235U DEL BANCO DE ENTRADA:3.5545
=====

          PARAMETROS DEL NUEVO BANCO

          GADOLINIO = 18.45
          URANIO    = 3.8945
          # DE CASOS = 18

=====

ESTÁN CORRECTOS LOS PARAMETROS DE ENTRADA (S/N)?
    
```

Figura A.38. Despliegue de los parámetros de entrada.

El despliegue de los datos de entrada arriba mencionado, se divide en cuatro partes, esto es de acuerdo a las pausas que aparecen en cada una de ellas; el primer letrero indica el inicio de la presentación; en segundo plano, los nombres de los Bancos de entrada utilizados en la interpolación llevada a cabo anteriormente y la cual servirá como base para el cálculo analizado; en tercer término, los identificadores para cada uno de los "casos" del Banco de salida; por último, los parámetros del Banco de entrada⁸, seguidos de los porcentajes de gadolinia y uranio del Banco de salida, así como el número de "casos" incluidos en éste. Cada una de las partes mencionadas va acompañada de la pausa correspondiente.

Es importante señalar tres cuestiones importantes; primero, que en esta parte del proceso no pueden modificar los datos desplegados de forma interactiva; segundo, el despliegue depende en gran medida de la terminal utilizada; por último, el orden en el que se muestran las Figuras es el mismo en el que aparecen los mensajes en pantalla. Posteriormente se abundará más sobre esta cuestión.

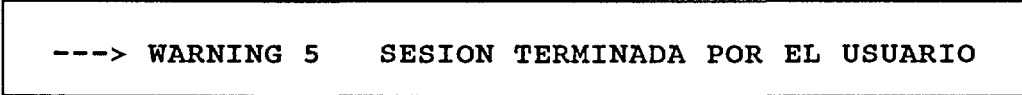
Al desplegar la última parte de la información de entrada, el sistema solicita una respuesta afirmativa (lo cual se logra tecleando una letra "S" mayúscula o minúscula), en caso de que la información desplegada sea la correcta. Si es el caso, se despliega el mensaje mostrado en la Figura A.39, indicando que se está procesando la información, esto es, se inicia el proceso de evaluación.



```
Un momento estoy procesando ....
```

Figura A.39. Información durante el proceso.

Si se considera que la información no es la correcta, esto es, se responde negativamente a la pregunta anterior (tecleando la letra "N" en mayúsculas o minúsculas, o cualquier otro carácter), el sistema concluye la ejecución del mismo, desplegando el mensaje mostrado en la Figura A.40. Para modificar alguno de los parámetros proporcionados es necesario volver a ejecutar desde el principio el presente módulo.



```
---> WARNING 5 SESION TERMINADA POR EL USUARIO
```

Figura A.40. Mensaje si la respuesta es negativa.

⁸ El Banco mencionado sirve de base para aquellos valores que no cambian en la interpolación.

Si el sistema no detecta ningún error durante la ejecución, termina desplegando el nombre del archivo obtenido durante el proceso de evaluación, como se muestra en la Figura A.41. El archivo mencionado se describe a continuación.

BANCO DE SALIDA : Es un archivo binario que contiene el Banco de Datos Nucleares generado mediante la evaluación de los polinomios interpolantes obtenidos previamente con una interpolación.

```
*****
*
*           EL PROGRAMA DE EVALUACION           *
*   UTILIZANDO LOS COEFICIENTES DE LOS         *
*   POLINOMIOS INTERPOLANTES                   *
*
*           FINALIZO SIN PROBLEMAS              *
*   !!!! VERIFIQUE SUS RESULTADOS !!!!         *
*
*   BANCO DE SALIDA : L1REM999.REC             *
*
*****
```

Figura A.41. Mensaje para indicar el fin de la ejecución.

A.3.1.- Errores del Módulo

Durante el proceso de ejecución pueden ocurrir errores que hagan que termine ésta, con lo cual los resultados no son satisfactorios. En esta sección se hace un resumen de aquellos errores que pueden ocurrir durante la ejecución y que no han sido mencionados hasta el momento, acompañado en cada caso del mensaje que despliega el sistema, una breve explicación de la causa y finalmente la posible solución.

```
--> ERROR 6   EL # DE CASOS DEL BANCO DE ENTRADA NO
              COINCIDE CON EL DEL BANCO DE SALIDA
```

Figura A.42. Mensaje de error.

El sistema detectó que el número de "casos" del Banco de salida no coincide con el número de "casos" del Banco de entrada. Verifique que el Banco proporcionado tenga el número de "casos" adecuado.

--> ERROR 8 EN LA LECTURA DEL BANCO DE ENTRADA

Figura A.43. Mensaje de error.

El archivo proporcionado como Banco de entrada contiene información distinta de la que realmente tiene un Banco. Verifique que el archivo proporcionado realmente sea un Banco de Datos Nucleares

**--> ERROR 9 EL BANCO DE SALIDA ES IGUAL (DE ACUERDO A LOS
% DE ²³⁵U Y Gd₂O₃) AL BANCO DE ENTRADA**

Figura A.44. Mensaje error.

Se proporcionaron porcentajes de uranio y gadolinia iguales a los que contiene el Banco de entrada. Verifique los porcentajes de uranio y gadolinia del Banco de entrada y ejecute nuevamente el sistema proporcionando los valores adecuados.

**--> ERROR 14 SE ENCONTRO LA MARCA DE FIN DE ARCHIVO EN
EL ARCHIVO DE COEFICIENTES**

Figura A.45. Mensaje de error.

El archivo de coeficientes proporcionado no es el que realmente contiene los coeficientes de los polinomios interpolantes para el caso analizado. Ejecute nuevamente el sistema proporcionando el archivo de coeficientes que corresponda a la interpolación solicitada.

Referencias

- [1] Torres C.; Alonso G.
Guía para documentar un Banco Nuclear de Datos de Laguna Verde con los códigos FMS de SCANDPOWER
Reporte IT.SN/DFR-123, pags. 2-3.
ININ, Junio de 1993.

- [2] Shardhamar T.; Halbritter J. P.
User Manual for RECORD, FMS Volume II
FMS/2/RECORD/B-01, pags. 10-11.
SCANDPOWER, Diciembre de 1989.

- [3] Glasstone S.; Sesonske A.
Ingeniería de Reactores Nucleares, pag. 166.
Editorial Reverté, 1968.

- [4] López H. A.; Esquivel J. L.
Generación de Bancos de Datos Nucleares por Interpolación
Reporte IIE/13/10046/1 04/P
IIE, Octubre de 1994.

- [5] Burden R. L.; Faires. J. D.
Análisis Numérico, pags. 99-109.
Grupo Editorial Iberoamérica, 1985.

- [6] Becker E.; Carey G.; Oden J.
Finite Elements, An Introduction, vol. I, pag. 161.
Prentice Hall, 1981.

- [7] Golub G; Van Loan Ch.
Matrix Computations, pag. 106.
Johns Hopkins University Press, 1990.

- [8] Forsythe G; Malcolm M.; Moler C.
Computer Methods for Mathematical Computations, pags. 41-50.
Prentice Hall, 1977.

- [9] Hernández H.
Generación de una base de datos para el código FCSII del sistema FMS
Reporte IT.ETN/DSN-280.
ININ, Noviembre de 1995.

Bibliografía

- Connolly T. J.
Fundamentos de Ingeniería Nuclear
Editorial LIMUSA, 1983.
- Glasstone S.; Sesonske A.
Ingeniería de Reactores Nucleares, pag. 166.
Editorial Reverté, 1968.
- Villar G. E.
Energía Atómica
Editorial Dossat, 1959.
- López H. A.; Esquivel J. L.
Generación de Bancos de Datos Nucleares por Interpolación
Reporte IIE/13/10046/I 04/P
IIE, Octubre de 1994.
- Burden R. L.; Faires. J. D.
Análisis Numérico
Grupo Editorial Iberoamérica, 1985.
- Allen R.; Pruess S; Shampine L.
Fundamentals of Numerical Computing
Serie Vínculos Matemáticos,
Facultad de Ciencias, U.N.A.M., 1986.
- Erick B.; Graham F.; Tinsley J.
FINITE ELEMENTS, An Introduction, vol. I
Prentice Hall, 1981.
- Zienkiewicz, O. C.
The Finite Element Method
McGraw Hill, 1977.
- Norrie D. H.; De Vries G.
An Introduction to Finite Element Analysis
Academic Press, 1978.

- Desai C.; Abel J.
INTRODUCTION TO THE FINITE ELEMENT METHOD,
a numerical method for engineering analysis
VAN NOSTRAND REINHOLD Eds., 1972.
- Gallagher R.
Finite Element Analysis, Fundamentals
Prentice Hall, 1975.
- Lewis P. E.; Ward J. P.
The Finite Element Method, Principles and Applications
Addison Wesley, 1991.
- Jin J.
The Finite Element Method in Electromagnetics
John Wiley & Sons, 1973.
- Segerlind L.
Applied Finite Element Analysis
John Wiley & Sons, 1976.
- Ciarlet P.
THE FINITE ELEMENT METHOD FOR ELLIPTIC PROBLEMS
North-Holland Publishing, 1978.
- Haaser B.; LaSalle J.; Sullivan J.
Análisis Matemático 2
Editorial Trillas, 1970.
- Griffel D. H.
Applied Functional Analysis
John Wiley & Sons, 1988.
- McAllister F.; Passow E.; Roullet J.
Algorithms for Computing Shape Preserving Spline Interpolations to Data
Mathematics of Computation, vol 3, number 139, jul 1977.
- VMS System Manager's Manual
Digital Equipment Corp., Junio de 1989.
- VAX FORTRAN, Language Reference Manual Vol. 2
Digital Equipment Corp., Junio de 1989.

- VMS Release Notes Vol. 2
Digital Equipment Corp., Noviembre de 1991.
- User Manual for RECORD 89-1ª, FMS Vol. II
SCANDPOWER A/S, Diciembre de 1989.
- Forsythe G; Malcolm M.; Moler C.
Computer Methods for Mathematical Computations
Prentice Hall, 1977.
- Golub. G; Van Loan Ch.
Matrix Computations, pag. 110.
Johns Hopkins University Press, 1990.
- Kahaner D.; Moler C.; Nash S.
Numerical Methods and Software
Prentice Hall, 1989.
- Hernández H.
Programa Extractor de Datos de RECORD POSTREC Ver. 1.0
Reporte IT.GSN/DFR-207
ININ, Mayo de 1994.
- Torres C.; Alonso G.
Guía para documentar un Banco Nuclear de Datos de Laguna Verde con los
códigos FMS de SCANDPOWER
Reporte IT.SN/DFR-123
ININ, Junio de 1993.
- MATLAB User's Guide
MATH Works Inc., 1994.
- Hanselman D.; Littlefield.
Mastering MATLAB 5, A Comprehensive Tutorial and Reference
Prentice Hall, 1998.