

**Centro de Investigación Científica y de
Educación Superior de Ensenada**



**UN METODO DE REGULARIZACION PARA
ESTIMAR FUNCIONES BIVARIANTES**

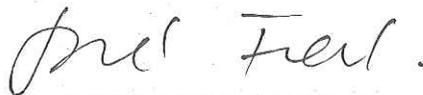
**TESIS
MAESTRIA EN CIENCIAS**

ROBERTO NUNEZ GONZALEZ

Ensenada, B. Cfa. Mexico.

Febrero de 2000

TESIS DEFENDIDA POR
Roberto Núñez González
Y APROBADA POR EL SIGUIENTE COMITÉ



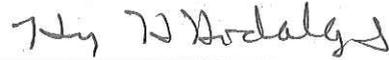
M.C. José Douglas Frez Cárdenas

Director del Comité



Dr. Francisco Javier Esparza Hernández

Miembro del Comité



Dr. Hugo Homero Hidalgo Silva

Miembro del Comité



Dr. José Gómez Valdés

Miembro del Comité



Dr. Jesús Favela Vara

*Jefe del Departamento de Ciencias de
la Computación*



Dr. Federico Graef Ziehl

Director de Estudios de Posgrado

16 de febrero del 2000

C. I. C. E. S. E.

CENTRO DE INVESTIGACION CIENTIFICA Y DE EDUCACION

SUPERIOR DE ENSENADA

DIVISION DE FISICA APLICADA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

**UN METODO DE REGULARIZACION PARA ESTIMAR
FUNCIONES BIVARIANTES**

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el

grado de **MAESTRO EN CIENCIAS** presenta:

ROBERTO NÚÑEZ GONZÁLEZ

Ensenada, Baja California, México. Febrero de 2000

RESUMEN de la Tesis de **ROBERTO NÚÑEZ GONZÁLEZ**, presentada como requisito parcial, para la obtención del grado de **MAESTRO EN CIENCIAS en CIENCIAS DE LA COMPUTACION**. Ensenada, Baja California, México. Febrero de 2000.

UN METODO DE REGULARIZACION PARA LA ESTIMACION DE FUNCIONES BIVARIANTES

Resumen aprobado por:



Dr. José Frez C.
Director de Tesis

Se aplica un método de regularización para resolver el problema inverso de una ecuación de Fredholm de primera clase en dos dimensiones. Como resultado se obtiene una expresión compacta para el estimador de la función incógnita. Esta expresión implica diversas dificultades, incluyendo el cálculo de integrales dobles y cuádruples.

Con el procedimiento computacional resultante, se resuelven dos problemas inversos con solución analítica dada. En uno de ellos el problema directo tiene solución analítica y valida el procedimiento computacional; en el otro, el problema directo se resuelve numéricamente y presenta mayores retos al procedimiento computacional. Ambos son resueltos agregando o no ruido (gaussiano) a la función que describe las observaciones. Para cada estimación, se obtienen los residuales de las observaciones y los errores del modelo estimado.

Los programas son optimizados para reducir los tiempos de cómputo. Primero, son optimizados mediante análisis de ciclos; después, se aplica una optimización automática usando opciones del compilador y, por último, se insertan directivas para una paralelización explícita del código fuente.

La limitante del procedimiento computacional es el uso fuerte de recursos computacionales, esencialmente de CPU. Se plantean actividades futuras que tienden a resolver esta limitante.

Palabras Clave: Inversión, Métodos de Regularización, Ecuación de Fredholm de primera clase, Métodos numéricos, Optimización, Paralelización.

ABSTRACT of the Thesis of **ROBERTO NÚÑEZ GONZÁLEZ**, presented as partial requirement to obtain the **MASTER IN SCIENCE** grade in **COMPUTER SCIENCE**. Ensenada, Baja California, México. February 2000.

A REGULARIZATION METHOD FOR ESTIMATING TWO DIMENSIONAL FUNCTIONS

ABSTRACT

A regularization method is developed to solve the inverse problem of a Fredholm equation of first kind in two dimensions. As result, a compact estimator is obtained.

The resulting computational procedure es utilized for solving two inverse problems, both with given analytic solution. One of them allows full analytic calculations and is used mainly for validating the procedure. For the second one, the forward problem is numerically solved and constitutes a more strong test to our procedure. Additional gaussian noise is optionally added to the function that describes the observations. For each estimation, observation residuals and estimated model error are calculated.

The programs are optimized to reduce computer time. First, loops are analyzed; next, the code is also optimized with the use of compiler options; finally, directives are introduced to the code for explicit parallelization.

The computational procedure is essentiality limited by a strong use of CPU time. Alternatives to solve this limitation are proposed consequently.

Keywords: Inversion, Regularization methods, Fredholm equation of the first kind, Numerical methods, Optimization, Parallelization.

DEDICATORIA

A mi esposa Gabby y a mi hija Athena Vianney
por su gran amor y cariño.
Sin ellas no hubiese podido concluir este trabajo.

AGRADECIMIENTOS

A Dios, por permitirme llegar hasta aquí.

A mi Madre y Hermanos, por el esfuerzo que hicieron por mí y por su gran cariño.

A la Familia Molina Medina, que me han apoyado como a su hijo y hermano en todo momento y de manera incondicional.

A mi asesor de tesis, Dr. José Frez C. por sus críticas constructivas, sugerencias y apoyo, fundamentales para el término de este trabajo.

Al Comité de tesis, Dr. Francisco Esparza, Dr. Hugo Hidalgo y Dr. José Gómez, por sus valiosos comentarios y sugerencias.

Al grupo de maestros de Ciencias de la Computación, por sus enseñanzas.

A mis compañeros de estudios, gracias a los cuales me impulse a mejorar.

A Ruth Eaton y Jaime Carlos Villegas, por las facilidades que me brindaron para realizar mi trabajo de tesis.

Al Centro de Cómputo de CICESE, por el uso de las instalaciones de cómputo, en especial a Julian Delgado.

Al Departamento de Servicios Escolares.

A Ivonne Best, por su paciencia y apoyo.

Al CONACYT por su apoyo económico.

Contenido

I	Antecedentes	1
I.1	Introducción	1
I.2	Antecedentes. Método de Regularización.	3
I.2.1	Método de Mínimos Cuadrados Estándar	7
I.2.2	Método de Mínimos Cuadrados Ponderados	8
I.2.3	Método de Mínima Longitud	9
I.2.4	Métodos de Regularización	10
I.2.5	Análisis de Residuales	14
I.3	Desarrollo de un Método de Regularización, Caso Bivariante	16
I.4	Objetivo de la Tesis	23
II	Procedimiento de Inversión. Validación del Procedimiento Numérico	
	Computacional	25
II.1	Definición del Problema	27
II.2	Solución Analítica del Problema Directo	30
II.3	Planteamiento del Problema Inverso	31

II.4	Métodos de Integración Numérica	34
II.5	Cálculo Numérico de Integrales	38
II.5.1	Cálculo de $e_i(r)$	38
II.5.2	Cálculo de S	45
II.6	Generación del Vector de Datos	48
II.7	Solución del Problema Inverso	54
II.8	Análisis de Resultados. Conclusiones	84
III Aplicación del Procedimiento de Inversión		86
III.1	Definición del Problema	86
III.2	Planteamiento del Problema Directo	89
III.3	Planteamiento del Problema Inverso	90
III.4	Generación del Vector de Datos	91
III.5	Solución del Problema Inverso	96
III.6	Análisis de Resultados. Conclusiones	123
IV Programas Computacionales		125
IV.1	Características de la Computadora	125
IV.2	Algoritmos y Codificación	126
IV.2.1	Programa directo.c	128
IV.2.2	Programa matriz_s.c	132
IV.2.3	Programa invertir.c	138
IV.3	Optimización	143

IV.4 Conclusiones	149
V Conclusiones y Perspectivas	151
V.1 Conclusiones	151
V.2 Perspectivas	155
Literatura Citada	156
Apéndice A	160

Lista de Figuras

Figura	Página
1 Gráfica de una Función Gaussiana bivalente: $x_i = 0, y_i = 0, l_x = 1.0$ y $l_y = 1.0$	27
2 Gráfica de la función $m(x, y) = a + bx + cy + dxy$, con parámetros $a = b = c = d = 1.0$ en la región de interés	28
3 Gráfica de la función g , con parámetros $l_x = l_y = 1.0$ y $a = b = c = d = 1.0$, sobre la región donde se obtienen las observaciones.	31
4 Gráfica de los puntos (x_i, y_i) utilizados en el problema de validación	50
5 Gráfica de $\hat{m}(r)$ para datos sin ruido; $\alpha = 1.0$	54
6 Gráfica de residuales para datos sin ruido; $\alpha = 1.0$	54
7 Gráfica del error para datos sin ruido	55
8 Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 5.0% de error	56
9 Gráfica de residuales para $\alpha = 0.00001$; 5.0% de error	56
10 Gráfica del error para $\alpha = 0.00001$; 5.0% de error	57
11 Gráfica de $\hat{m}(r)$ para $\alpha = 0.001$; 5.0% de error	57
12 Gráfica de residuales para $\alpha = 0.001$; 5.0% de error	58

13	Gráfica del error para $\alpha = 0.001$; 5.0% de error	58
14	Gráfica de $\hat{m}(r)$ para $\alpha = 0.01$; 5.0% de error	59
15	Gráfica de residuales para $\alpha = 0.01$; 5.0% de error	59
16	Gráfica del error para $\alpha = 0.01$; 5.0% de error	60
17	Gráfica de $\hat{m}(r)$ para $\alpha = 0.1$; 5.0% de error	60
18	Gráfica de residuales para $\alpha = 0.1$; 5.0% de error	61
19	Gráfica del error para $\alpha = 0.1$; 5.0% de error	61
20	Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 5.0% de error	62
21	Gráfica de residuales para $\alpha = 1.0$; 5.0% de error	62
22	Gráfica del error para $\alpha = 1.0$; 5.0% de error	63
23	Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 5.0% de error	63
24	Gráfica de residuales para $\alpha = 10.0$; 5.0% de error	64
25	Gráfica del error para $\alpha = 10.0$; 5.0% de error	64
26	Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 5.0% de error	65
27	Gráfica de residuales para $\alpha = 100.0$; 5.0% de error	65
28	Gráfica del error para $\alpha = 100.0$; 5.0% de error	66
29	Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 1.0% de error	67
30	Gráfica de residuales para $\alpha = 0.00001$; 1.0% de error	67
31	Gráfica del error para $\alpha = 0.00001$; 1.0% de error	68
32	Gráfica de $\hat{m}(r)$ para $\alpha = 0.01$; 1.0% de error	68
33	Gráfica de residuales para $\alpha = 0.01$; 1.0% de error	69
34	Gráfica del error para $\alpha = 0.01$; 1.0% de error	69

35	Gráfica de $\hat{m}(r)$ para $\alpha = 0.1$; 1.0% de error	70
36	Gráfica de residuales para $\alpha = 0.1$; 1.0% de error	70
37	Gráfica del error para $\alpha = 0.1$; 1.0% de error	71
38	Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 1.0% de error	71
39	Gráfica de residuales para $\alpha = 1.0$; 1.0% de error	72
40	Gráfica del error para $\alpha = 1.0$; 1.0% de error	72
41	Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 1.0% de error	73
42	Gráfica de residuales para $\alpha = 10.0$; 1.0% de error	73
43	Gráfica del error para $\alpha = 10.0$; 1.0% de error	74
44	Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 1.0% de error	74
45	Gráfica de residuales para $\alpha = 100.0$; 1.0% de error	75
46	Gráfica del error para $\alpha = 100.0$; 1.0% de error	75
47	Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 0.2% de error	76
48	Gráfica de residuales para $\alpha = 0.00001$; 0.2% de error	76
49	Gráfica del error para $\alpha = 0.00001$; 0.2% de error	77
50	Gráfica de $\hat{m}(r)$ para $\alpha = 0.1$; 0.2% de error	77
51	Gráfica de residuales para $\alpha = 0.1$; 0.2% de error	78
52	Gráfica del error para $\alpha = 0.1$; 0.2% de error	78
53	Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 0.2% de error	79
54	Gráfica de residuales para $\alpha = 1.0$; 0.2% de error	79
55	Gráfica del error para $\alpha = 1.0$; 0.2% de error	80
56	Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 0.2% de error	80

57	Gráfica de residuales para $\alpha = 10.0$; 0.2% de error	81
58	Gráfica del error para $\alpha = 10.0$; 0.2% de error	81
59	Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 0.2% de error	82
60	Gráfica de residuales para $\alpha = 100.0$; 0.2% de error	82
61	Gráfica del error para $\alpha = 100.0$; 0.2% de error	83
62	Gráfica de la función gaussiana (44): $x_i = 0, y_i = 0$	87
63	Gráfica de la función $m(x, y)$ utilizada en el segundo problema	88
64	Gráfica de los puntos (x_i, y_i) utilizados en el problema de aplicación	92
65	Gráfica de $\hat{m}(r)$ para datos sin ruido	96
66	Gráfica de residuales para datos sin ruido	96
67	Gráfica del error para datos sin ruido	97
68	Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 5.0% de error	98
69	Gráfica de residuales para $\alpha = 0.00001$; 5.0% de error	98
70	Gráfica del error para $\alpha = 0.00001$; 5.0% de error	99
71	Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 5.0% de error	99
72	Gráfica de residuales para $\alpha = 1.0$; 5.0% de error	100
73	Gráfica del error para $\alpha = 1.0$; 5.0% de error	100
74	Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 5.0% de error	101
75	Gráfica de residuales para $\alpha = 10.0$; 5.0% de error	101
76	Gráfica del error para $\alpha = 10.0$; 5.0% de error	102
77	Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 5.0% de error	102

78	Gráfica de residuales para $\alpha = 100.0$; 5.0% de error	103
79	Gráfica del error para $\alpha = 100.0$; 5.0% de error	103
80	Gráfica de $\hat{m}(r)$ para $\alpha = 1000.0$; 5.0% de error	104
81	Gráfica de residuales para $\alpha = 1000.0$; 5.0% de error	104
82	Gráfica del error para $\alpha = 1000.0$; 5.0% de error	105
83	Gráfica de $\hat{m}(r)$ para $\alpha = 10000.0$; 5.0% de error	105
84	Gráfica de residuales para $\alpha = 10000.0$; 5.0% de error	106
85	Gráfica del error para $\alpha = 10000.0$; 5.0% de error	106
86	Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 1.0% de error	107
87	Gráfica de residuales para $\alpha = 0.00001$; 1.0% de error	107
88	Gráfica del error para $\alpha = 0.00001$; 1.0% de error	108
89	Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 1.0% de error	108
90	Gráfica de residuales para $\alpha = 10.0$; 1.0% de error	109
91	Gráfica del error para $\alpha = 10.0$; 1.0% de error	109
92	Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 1.0% de error	110
93	Gráfica de residuales para $\alpha = 100.0$; 1.0% de error	110
94	Gráfica del error para $\alpha = 100.0$; 1.0% de error	111
95	Gráfica de $\hat{m}(r)$ para $\alpha = 1000.0$; 1.0% de error	111
96	Gráfica de residuales para $\alpha = 1000.0$; 1.0% de error	112
97	Gráfica del error para $\alpha = 1000.0$; 1.0% de error	112
98	Gráfica de $\hat{m}(r)$ para $\alpha = 10000.0$; 1.0% de error	113
99	Gráfica de residuales para $\alpha = 10000.0$; 1.0% de error	113

100	Gráfica del error para $\alpha = 10000.0$; 1.0% de error	114
101	Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 0.2% de error	115
102	Gráfica de residuales para $\alpha = 0.00001$; 0.2% de error	115
103	Gráfica del error para $\alpha = 0.00001$; 0.2% de error	116
104	Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 0.2% de error	116
105	Gráfica de residuales para $\alpha = 10.0$; 0.2% de error	117
106	Gráfica del error para $\alpha = 10.0$; 0.2% de error	117
107	Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 0.2% de error	118
108	Gráfica de residuales para $\alpha = 100.0$; 0.2% de error	118
109	Gráfica del error para $\alpha = 100.0$; 0.2% de error	119
110	Gráfica de $\hat{m}(r)$ para $\alpha = 1000.0$; 0.2% de error	119
111	Gráfica de residuales para $\alpha = 1000.0$; 0.2% de error	120
112	Gráfica del error para $\alpha = 1000.0$; 0.2% de error	120
113	Gráfica de $\hat{m}(r)$ para $\alpha = 10000.0$; 0.2% de error	121
114	Gráfica de residuales para $\alpha = 10000.0$; 0.2% de error	121
115	Gráfica del error para $\alpha = 10000.0$; 0.2% de error	122

Lista de Tablas

Tabla	Pág.
I $e_i(x, y)$ para $(x_i, y_i) = (0, 0)$ y $(x, y) = (0, 0)$	43
II $e_i(x, y)$ para $(x_i, y_i) = (0.2, 0.2)$ y $(x, y) = (0, 0)$	43
III $e_i(x, y)$ para $(x_i, y_i) = (-0.2, -0.2)$ y $(x, y) = (0, 0)$	44
IV $e_i(x, y)$ para $(x_i, y_i) = (5.0, -3.0)$ y $(x, y) = (1.0, -3.0)$	44
V Puntos de observación para el problema de validación	50
VI Observaciones sin ruido utilizadas en el problema de validación	51
VII Observaciones con ruido para el problema de validación	53
VIII Puntos (x_i, y_i) utilizados en el problema de aplicación	91
IX Observaciones sin ruido para el problema de aplicación	93
X Observaciones con ruido para el problema de aplicación	95
XI Tiempos de ejecución en paralelo de <code>matriz_s.c</code>	148

Capítulo I

Antecedentes

I.1 Introducción

Una parte del proceso de investigación científica consiste en trabajar con conjuntos de observaciones (datos) y determinar cómo éstos se ajustan a un modelo teórico dado. Una labor más difícil consiste en inferir en forma (aproximadamente) unívoca, un modelo teórico adecuado a partir de las observaciones. Para medir el ajuste, se escoge o diseña una *función criterio* (también llamada función objeto o mérito) que evalúa comúnmente una “distancia” entre los datos y el modelo, este último con una opción particular de parámetros. La meta es determinar el conjunto de parámetros del modelo que llevan a un valor óptimo de la función criterio. Una vez resuelto el problema de optimización, se obtienen ecuaciones para determinar los parámetros de mejor ajuste. Una ecuación de éstas se llama *estimador*.

Debido a que las observaciones están sujetas a errores de medición (ruido, en el

contexto de procesamiento de señales), éstas no pueden ajustar exactamente al modelo estimado, aún cuando éste es correcto. Así, para determinar si el modelo estimado es apropiado o no, se mide la bondad del ajuste mediante métodos estadísticos.

En este trabajo de tesis se resuelve el siguiente problema. Dada una ecuación de Fredholm de Primera Clase en dos dimensiones (2-D)

$$g(r) = \int K(r, r') m(r') dr'$$

se estima la función bivalente $m(r')$ a partir de la función $g(r)$ que se conoce en varios puntos r y de la función $K(r, r')$ (llamada kernel de la ecuación). Este es un *Problema Inverso*, mientras que el *Problema Directo* consiste en calcular $g(r)$ dados $K(r, r')$ y $m(r')$. Dado que el problema inverso no tiene solución analítica general (ver más adelante) se diseña un procedimiento numérico computacional para encontrar una estimación óptimo según una función criterio a elegir. La variable r' representa un punto en el plano cartesiano y la integral indicada representa una integral doble.

Es común que cuando se resuelven numéricamente problemas definidos en dos dimensiones éstos se reducen a problemas unidimensionales ordenando las matrices de datos e incógnitas por filas o columnas y convirtiéndolas, por lo tanto, en vectores que se relacionan, en el caso lineal, por una matriz.

El tratamiento directo de problemas inversos para estimar funciones bivariantes sin realizar la reducción mencionada tiene numerosas dificultades. Algunos sistemas comerciales resuelven problemas inversos en 2-D sin reducir su dimensionalidad, pero los detalles del desarrollo, tanto matemático como computacional, no están disponibles

fácilmente (Swanger *et al*, 1983).

En esta tesis se desarrolla, comprueba y aplica un método que resuelve problemas inversos en 2-D si reducirlos a una dimensión. En la sección I.3, se hacen las derivaciones matemáticas, mientras que en los Capítulos II y III se comprueba el procedimiento numérico computacional correspondiente. En el Capítulo IV, se hace una descripción de los programas computacionales, así como el proceso de optimización de los mismos incluyendo información acerca de tiempos de ejecución.

I.2 Antecedentes. Método de Regularización.

Los problemas que surgen en distintos campos de las ciencias se representan comúnmente mediante un modelo matemático en variables continuas pero, debido a la dificultad que existe para obtener una solución analítica o a la falta de ésta, se procede a discretizar el modelo y a resolverlo numéricamente en una computadora digital.

En el campo de la Física muchos problemas se modelan mediante ecuaciones diferenciales o ecuaciones integrales, siendo estas últimas de interés en esta tesis. Las ecuaciones integrales se dividen en cuatro tipos, ecuaciones de Fredholm y de Volterra, que pueden ser, cada una de ellas, de primera clase o segunda clase (Arfken, 1985).

Nuestro interés se centra en la ecuación de Fredholm de Primera Clase, que para el caso univariante (1-D) se representa del siguiente modo

$$y(s) = \int_a^b K(s,t) x(t) dt \quad (1)$$

donde los límites de integración a y b son fijos, y la función $K(s, t)$ es el kernel de la ecuación.

El problema de determinar la función $y(s)$ dados el kernel $K(s, t)$ y la función $x(t)$, se le conoce como *Problema Directo* y, en este caso, sólo se necesita realizar una integración, ya sea analítica o numérica.

Al problema de determinar la función $x(t)$ si se conocen $K(s, t)$ y $y(s)$ se le conoce como *Problema Inverso*. El problema inverso definido mediante la ecuación (1) presenta por lo menos dos dificultades suponiendo que, por lo menos, existe una solución. En primer lugar, es *inestable* ya que pequeñas variaciones en la función $y(s)$ se pueden traducir en grandes variaciones en la estimación de la función $x(t)$. En segundo lugar, dado que el operador integral tiene un espacio nulo no vacío, el problema inverso puede tener un número infinito de soluciones (el espacio nulo se define por todos aquellos x_0 tales que $\int_a^b K(s, t) x_0(t) dt = 0$). Estas dos características (inestabilidad y no-unicidad) hacen que el problema inverso de una ecuación de Fredholm de primera clase sea un *Problema Inverso Mal Condicionado* (Craig y Brown, 1986). No existen métodos analíticos generales para resolverlo.

Un tratamiento matemático adecuado de la situación física estudiada garantiza la existencia de una solución de un problema inverso.

La función $K(s, t)$ usualmente representa la respuesta de un sistema físico. Esta función se obtiene analíticamente o por experimentos controlados de laboratorio.

En un contexto experimental, la función $y(s)$ de la ecuación (1) representa datos experimentales, obtenidos en un rango finito de argumentos s_i . Con frecuencia, estos

puntos son pocos y los datos contienen errores. Finalmente, la función $x(t)$ es la función a determinar (función incógnita) en el problema inverso. Obviamente es deseable que exista la mayor cantidad de información acerca de esta función, que se transmite desde los valores de $y(s_i)$. Bajo estas consideraciones, la ecuación (1) es discreta en los datos y continua en las incógnitas

$$y_i = \int_a^b K_i(t) x(t) dt \quad (2)$$

donde $y_i = y(s_i)$ y $K_i(t) = K(s_i, t)$. Nuevamente no existen métodos analíticos generales para resolver esta ecuación por lo que es usual discretizar el argumento continuo de la ecuación y aplicar métodos numéricos para resolverla. De esta forma la ecuación (2) es discreta en los datos y discreta en las incógnitas

$$y = A x \quad (3)$$

donde y es el vector de datos, A es una matriz que representa al operador integral y x es el vector de incógnitas. Esta ecuación puede asimilarse al caso lineal originalmente discreto que tiene la misma forma, pero con el cual no debe confundirse porque, por lo ya dicho, conlleva dificultades muy específicas. Suponiendo independencia lineal entre las diferentes filas (columnas) de la matriz A , el sistema (3) representa un sistema de ecuaciones que puede ser *bien determinado* (el número de ecuaciones es igual al número de incógnitas) y tiene solución única; *sobredeterminado* (mayor número de ecuaciones que de incógnitas) que no tiene solución, o *indeterminado* (mayor número de incógnitas

que de ecuaciones) que tiene un número infinito de soluciones (Hager, 1988).

Una solución del problema lineal algebraico (3) se puede plantear como

$$x = A^{-g}y$$

donde A^{-g} es la *matriz pseudo-inversa* de A . Si la matriz A es cuadrada y no singular

$$A^{-g} = A^{-1}$$

Si el sistema de ecuaciones (3) se genera a partir de la discretización del problema inverso mal condicionado (1), el sistema de ecuaciones es intrínsecamente indeterminado. En la práctica, cuando se resuelve el sistema (3) en una computadora, aplicamos algún criterio de estimación óptima para resolver el sistema de ecuaciones.

Algunos de los criterios elementales (que aplican criterios de optimización) que se utilizan para obtener una solución del sistema de ecuaciones (3), cuando es sobre o indeterminado, son: el Método de Mínimos Cuadrados Estándar, el de Mínimos Cuadrados Ponderados y el Método de Mínima Longitud. Cuando estos procedimientos producen sistemas indeterminados se utilizan los Métodos de Regularización (Tikhonov, 1977) o de Descomposición Espectral (Lanczos, 1961).

Los métodos mencionados anteriormente se describen a continuación, excepto los de Descomposición Espectral que no utilizamos, ni analíticamente ni algorítmicamente, en esta tesis.

I.2.1 Método de Mínimos Cuadrados Estándar

Si se conoce que el sistema de ecuaciones es sobredeterminado, formalmente no tiene solución, por lo cual se busca una solución que cumpla un cierto criterio. El método más popular aplica el criterio de minimizar la norma cuadrática de los residuales entre los datos y los datos predichos por la solución x , esto es

$$\min \{ \|y - Ax\|^2 \} \quad (4)$$

donde $\|z\|^2 = z^T z$ es la norma cuadrática o Euclidiana del vector z , siendo z^T el vector traspuesto de z . Este método supone que la sobredeterminación (incompatibilidad) resulta de la existencia de errores pequeños, aleatorios con promedio nulo y aditivos. Este criterio produce la siguiente solución analítica compacta (Hager, 1988)

$$\hat{x}_{mce} = (A^T A)^{-1} A^T y \quad (5)$$

donde \hat{x}_{mce} es el valor estimado. El símbolo $\hat{}$ indica valor estimado.

Sin embargo, si el sistema (3) es esencialmente indeterminado, la matriz $A^T A$ es singular (cuasi-singular en la computadora). Suponiendo que se logre calcular $(A^T A)^{-1}$ en la computadora, el estimador produce sobreajuste, lo que se traduce en soluciones altamente oscilantes donde la componente de ruido no se restringe a estar representada en los residuales sino también, y altamente amplificada, en la solución (Frez, 1999).

I.2.2 Método de Mínimos Cuadrados Ponderados

Este método es una extensión del método de mínimos cuadrados estándar, ponderando las mediciones o datos gracias a un criterio auxiliar. El criterio más popular (pero no el único) consiste en darle más peso o importancia a los datos más exactos; esta información se incluye a través de la matriz de covarianza de los errores de observación C_{yy} , la cual, en el caso más simple y comúnmente usado, se construye como una matriz diagonal cuyos elementos son los cuadrados de los errores correspondientes a los valores y_i . El criterio que utiliza este método es

$$\min \left\{ \|(y - Ax)\|_{C_{yy}}^2 \right\} \equiv \min \left\{ (y - Ax)^T C_{yy}^{-1} (y - Ax) \right\}$$

donde se utiliza la nomenclatura $\|z\|_C^2 = z^T C z$. Se supone que se tienen observaciones con ruido

$$y = Ax + e$$

donde e es un vector aleatorio con media cero y covarianza C_{yy} , la cual se supone positiva definida. El criterio es una generalización de

$$\min \left\{ \sum_i \left(\frac{y_i - \sum_j a_{ij} x_j}{e_i} \right)^2 \right\}$$

donde se pondera el ajuste de cada dato con la medida de precisión $\frac{1}{e_i}$ correspondiente.

El estimador que satisface este criterio es

$$\hat{x}_{mcp} = (A^T C_{yy}^{-1} A)^{-1} A^T C_{yy}^{-1} y \quad (6)$$

que se reduce al estimador de mínimos cuadrados estándar cuando $C_{yy}^{-1} = I$, la matriz identidad. Este estimador se puede reescribir como

$$\hat{x}_{mcp} = H_{mcp} y$$

donde

$$H_{mcp} = (A^T C_{yy}^{-1} A)^{-1} A^T C_{yy}^{-1}$$

es la matriz pseudoinversa.

I.2.3 Método de Mínima Longitud

Si el sistema (3) es explícitamente indeterminado, entonces tiene un número infinito de soluciones, pero aún así se puede seleccionar de ese número infinito la más cercana a otra solución x_0 que nos parezca buena. El criterio de mínima longitud consiste en minimizar la norma cuadrática de la distancia que existe entre una solución x y una estimación inicial de la solución \hat{x}_0 , esto es

$$\min \{ \|x - \hat{x}_0\|^2 \} \quad (7)$$

de tal modo que

$$y = A x$$

Utilizando el método de coeficientes de Lagrange (Leithold, 1992) lo anterior se cumple resolviendo

$$\min \{ \|x - \hat{x}_0\|^2 + 2\lambda^T(y - Ax) \} \quad (8)$$

donde λ es un vector de coeficientes de Lagrange. Si suponemos que $\hat{x}_0 = 0$ el estimador resultante es (Hager, 1988)

$$\hat{x}_{ml} = A^T(A A^T)^{-1} y$$

de lo contrario

$$\hat{x}_{ml} = \hat{x}_0 + A^T(A A^T)^{-1} (y - A\hat{x}_0) \quad (9)$$

Un aspecto a considerar es que este estimador depende de la selección de \hat{x}_0 , por lo que una condición necesaria pero no suficiente es que la solución estimada \hat{x} es buena (en el sentido de resolver el problema físico) mientras \hat{x}_0 sea lo bastante buena.

I.2.4 Métodos de Regularización

Como se mencionó anteriormente, si el sistema de ecuaciones (3) es mal condicionado, los tres métodos anteriores fallan, ya que, en general, las matrices $A^T A$, $A^T C_{yy}^{-1} A$ y $A A^T$ son singulares (o cuasi-singulares).

La idea básica de los métodos de regularización es complementar algunos de los criterios anteriores con una o más condiciones adicionales que, al agregar información sobre el vector o función incógnita, estabilice o regularice la estimación (Thikonov y Arsenin, 1977, Frez, 1999). Así, un criterio más general de optimización es

$$\min \left\{ \|y - Ax\|_{C_{yy}^{-1}}^2 + \alpha \Omega(x) \right\} \quad (10)$$

donde Ω es la función regularizante y α es un parámetro global de regularización, o simplemente *parámetro de regularización*; este valor indica el peso relativo entre el ajuste de los datos y el filtro que estabiliza la estimación. Así, el estimador se puede escribir como $\hat{x}(\alpha)$.

En la estimación para el caso discreto, un operador regularizante que se usa comúnmente consiste en la norma de la diferencia entre el vector de la solución óptima x y otra \hat{x}_0 que se considera suficientemente buena, esto es

$$\Omega(x) = \|x - \hat{x}_0\|^2$$

así el criterio de optimización es

$$\min \left\{ \|y - Ax\|_{C_{yy}^{-1}}^2 + \alpha \|x - \hat{x}_0\|^2 \right\} \quad (11)$$

llamándose criterio de *Mínimos Cuadrados Estabilizado* (llamado también Damped Least Square o Ridge Regression Estimation en la literatura (Lawson y Hanson, 1974)).

Resolviendo la minimización, el estimador resulta

$$\hat{x} = \hat{x}_0 + (A^T C_{yy}^{-1} A + \alpha I)^{-1} A^T C_{yy}^{-1} (y - A\hat{x}_0) \quad (12)$$

Otra opción que se usa con frecuencia es que el operador regularizante agregue condiciones de suavización; en particular, puede ser la norma de algún operador dife-

rencial aplicado sobre la función solución. En los casos más simples el criterio utilizado es

$$\min \left\{ \|y - Ax\|_{C_{yy}^{-1}}^2 + \alpha \|D_p x\|^2 \right\} \quad (13)$$

donde D_p representa el operador derivada de orden p , discretizado y formando una matriz, que se aplica sobre el vector de incógnitas. Resolviendo el criterio anterior, el estimador resulta

$$\hat{x} = \left(A^T C_{yy}^{-1} A + \alpha D_{2p} \right)^{-1} A^T C_{yy}^{-1} y$$

donde D_{2p} es una matriz formada a partir de la discretización del operador derivada de orden $2p$ (Frez, 1999). Considerando la propiedad de que $D_{2p} = D_p^T D_p$, el estimador se reescribe como

$$\hat{x} = D_p^{-1} \left[D_p^{-T} A^T C_{yy}^{-1} A D_p^{-1} + \alpha I \right]^{-1} D_p^{-T} A^T C_{yy}^{-1} y \quad (14)$$

En la práctica, la inversión numérica del operador D_p puede producir dificultades, por lo que utilizamos *funciones de Green* para obtener el operador inverso de D_p (Frez, 1999). A continuación, se hace una breve descripción de las funciones de Green y como pueden utilizarse para representar el operador inverso de D_p .

Funciones de Green

Supongamos que tenemos el problema inverso

$$L x = f$$

donde L es un operador lineal autoadjunto que incluye condiciones de frontera homogéneas. La solución de este problema puede escribirse operacionalmente como

$$x = G f$$

donde G es la función de Green, que se calcula resolviendo (Greenberg, 1971, Arfken, 1985)

$$L G = \delta$$

donde δ de la función *delta de Dirac* (Arfken, 1985). Ya que δ representa un impulso ideal, G es también llamada *respuesta impulsiva* del sistema lineal L .

Suponiendo que L es un operador diferencial, la penúltima ecuación se escribe explícitamente como

$$x(s) = \int_a^b G(s,t) f(t) dt$$

Ahora, consideremos el caso (Frez, 1999)

$$D_p x(t) = z(t)$$

donde D_p es un operador diferencial de orden p ; la solución $x(t)$ se obtiene utilizando la función de Green G_p para el operador D_p con condiciones de frontera homogéneas

$$x(t) = \int G_p(t) z(t) dt$$

la cual discretizada queda

$$x = G_p S_x z$$

donde S_x es una matriz compuesta por los factores de integración y G_p es la representación discreta de la función de Green para el operador continuo D_p .

De lo anterior, se deduce que, para condiciones de frontera homogéneas para la función de Green

$$D_p^{-1} = G_p S_x \quad (15)$$

Por lo tanto, es posible utilizar una función de Green para obtener analíticamente el operador inverso del operador diferencial D_p . Esto evita tener que invertir numéricamente al operador D_p , evitándose posibles errores numéricos. Con este resultado, el estimador regularizado resulta

$$\hat{x} = G \left[G^T A^T C_{yy}^{-1} A G + \alpha I \right]^{-1} G^T A^T C_{yy}^{-1} y \quad (16)$$

donde $G = G_p S_x$. Para simplificar más este resultado, definimos la matriz $A_N = A G$, resultando (Frez, 1999)

$$\hat{x} = G \left[A_N^T C_{yy}^{-1} A_N + \alpha I \right]^{-1} A_N^T C_{yy}^{-1} y \quad (17)$$

I.2.5 Análisis de Residuales

Al utilizar los métodos de regularización para resolver problemas inversos, el procedimiento que se sigue es generar sistemáticamente un gran número de estimaciones \hat{x} ,

una para cada valor del parámetro de regularización. Un caso particular es cuando $\alpha = 0$ que corresponde a la solución de mínimos cuadrados.

Para cada estimación se construye el *vector de residuales* correspondiente

$$r = y - A\hat{x} \quad (18)$$

Una vez que se tienen varias estimaciones y sus correspondientes residuales, el paso siguiente es determinar cual de estas estimaciones es la más cercana a la solución real.

Un aspecto importante para determinar cual de las estimaciones obtenidas es la óptima, se hace un *análisis de los residuales*. Este análisis se basa en el hecho de que si suponemos que contamos con una buena estimación de la matriz de covarianza de los errores de los datos C_{yy} y de un buen modelo de los errores (gaussiano, con promedio nulo, etc.), podemos esperar de que el error del estimador sea pequeño y lo más importante, que las propiedades estadísticas del vector de residuales reflejen aproximadamente las del vector de errores de los datos.

Con las consideraciones anteriores se construye, para cada valor de α , la gráfica de los residuales normalizados con respecto de los errores estimados

$$\frac{r_i}{e_i} \text{ vs } i$$

donde r_i es el residual correspondiente al i -ésimo dato, e_i es el error estimado del i -ésimo dato e i corresponde al índice del dato. Lo normal es obtener una gráfica que oscile aleatoriamente alrededor de la línea cero con una amplitud que sobrepase los valores

1.0, 2.0 y 3.0 con una frecuencia del 65%, 95% y 99%, respectivamente.

La solución que se selecciona debe mostrar, además, “realismo” y estabilidad. Suponiendo que la solución no contiene fuertes oscilaciones, esto se mide por inspección directa del conjunto de soluciones $\hat{x}(\alpha)$.

I.3 Desarrollo de un Método de Regularización, Caso Bivariante

En el desarrollo que sigue, que forma el núcleo matemático de esta tesis, se generaliza el procedimiento desarrollado por *Yanovskaya y Ditmar*(1990) el cual resuelve un problema particular, la determinación del campo de velocidades mediante la inversión de tiempos de arribo de ondas que atraviezan un medio heterogéneo de velocidades.

Supongamos que los datos se toman en forma irregular dentro de una región del plano cartesiano. En este caso, es preferible plantear el problema semidiscreto

$$\int K_i(r)m(r)dr = g_i \quad (19)$$

donde r representa el vector de posición bidimensional, g es el vector de datos, $K_i(r)$ es el kernel de ecuación y $m(r)$ es la función a determinar. Los datos pertenecen también a una superficie bidimensional pero, para simplificar, se han reducido a un vector ordenándolos unidimensionalmente. La ecuación (19) supone que $m(r)$ varía continuamente en la coordenada r . La integración se hace sobre todo el espacio bidimensional y, en adelante, así se considera a menos que se indique otra cosa.

La estimación de la función incógnita $m(x, y)$ cumple las siguientes condiciones

1. La solución debe ser suave en el sentido de que

$$\min \left\{ \int_S |\nabla m(r)|^2 dr \right\}$$

donde $|\nabla m|$ es la magnitud del gradiente de $m(r)$

2. Sobre el contorno C_S del área S la *derivada normal exterior* se anula, esto es

$$\left(\frac{\partial m}{\partial n} \right)_{C_S} = 0$$

donde n es el vector normal al contorno C_S . En otras palabras $m(r)$ es constante en el contorno C_S .

Por lo tanto, buscamos una solución que minimice el funcional

$$(g - Km)^T C_{yy}^{-1} (g - Km) + \alpha \int_S |\nabla m(r)|^2 dr \quad (20)$$

donde α es el parámetro de regularización y usamos la notación para el operador K

$$(Km)_i = \int_S K_i(r) m(r) dr$$

donde $K_i(r)$ se debe entender como una función vectorial. Se supone que S es el plano completo (C_S se sitúa en el infinito) y $m(r)$ es constante en el infinito, de tal manera que $(\partial m / \partial n)_{C_S} = 0$.

Minimizando (20) (ver el Apéndice A)

se obtiene operacionalmente la siguiente ecuación integro-diferencial lineal en $m(r)$

$$\{K^T(r) C_{yy}^{-1} K + \alpha \nabla^2\} \hat{m} = K^T C_{yy}^{-1} g$$

o

$$\alpha \nabla^2 \hat{m} = K^T(r) C_{yy}^{-1} (K \hat{m} - g) \quad (21)$$

donde el símbolo de trasposición "T" se aplica sobre el carácter vectorial del operador integral K .

En el artículo de Yanovkaya y Ditmar (Yanovkaya y Ditmar, 1990) se obtiene una expresión similar a la ecuación (21), y en el mismo se propone como solución

$$\hat{m}(r) = \int F(r') G(r', r) dr' + N(r) \quad (22)$$

donde $F(r)$ es una función a determinar, $N(r)$ es una combinación lineal de funciones del espacio nulo del laplaciano ∇^2 y $G(r, r')$ es la función de Green para el laplaciano ∇^2 en 2-D (Arfken, 1985)

$$G(r', r) = (2\pi)^{-1} \ln |r' - r|^{-1} = G(r, r') \quad (23)$$

donde r y r' representan puntos en el plano cartesiano y $|r' - r|$ es la magnitud de la distancia entre los dos puntos.

En esta tesis se propone la misma solución y en los cálculos subsiguientes se demuestra que $F(r)$ puede ser determinada unívocamente satisfaciendo la ecuación

original, por lo que la expresión (22) es una buena solución de la ecuación (21).

Definiendo las siguientes funciones

$$e_i(r) = \int K_i(r')G(r', r)dr' \quad (24)$$

$$c_i = \int K_i(r)N(r)dr \quad (25)$$

$$u_i = \int e_i(r)F(r)dr \quad (26)$$

y sustituyendo la ecuación (22) en (21) se obtiene

$$\alpha \nabla^2 \left[\int dr' F(r')G(r', r) + N(r) \right] = \sum_i \left[K_i(r)C_{ii}^{-1} \left\{ \int dr'' K_i(r'') \left[\int dr' F(r')G(r', r'') + N(r'') \right] - g_i \right\} \right] \quad (27)$$

donde r' y r'' son variables de integración.

Analicemos el lado izquierdo de la igualdad (27). Ya que el operador ∇^2 actúa sobre la variable r resulta, al aplicar la definición de la función de Green (Arfken, 1985)

$$\alpha \nabla^2 \left[\int dr' F(r')G(r', r) + N(r) \right] = -\alpha \left[F(r) + \nabla^2 N(r) \right]$$

Como $N(r)$ representa una combinación lineal de funciones del espacio nulo de ∇^2 tenemos que $\nabla^2 N(r) = 0$. Por lo tanto, el lado izquierdo de la igualdad se reduce a

$$-\alpha F(r) \quad (28)$$

Analicemos el lado derecho de (27). Si cambiamos el orden de integración y utilizando las definiciones (24),(25) y (26) tenemos

$$\sum_i K_i(r) C_{ii}^{-1} \left[\int dr' F(r') \int dr'' K_i(r'') G(r', r'') + \int dr'' K_i(r'') N(r'') - g_i \right] =$$

$$\sum_i K_i(r) C_{ii}^{-1} \left[\int dr' F(r') e_i(r') + c_i - g_i \right] =$$

$$\sum_i K_i(r) C_{ii}^{-1} [u_i + c_i - g_i]$$

donde la expresión dentro del paréntesis no depende de r . Con esto, la igualdad queda

$$-\alpha F(r) = \sum_i K_i(r) C_{ii}^{-1} [u_i + c_i - g_i]$$

ó

$$\alpha F(r) = \sum_i K_i(r) C_{ii}^{-1} [g_i - u_i - c_i] \quad (29)$$

Multiplicando esta expresión por $e_j(r)$ e integrando sobre todo el espacio, se obtiene una expresión con respecto a u y c .

$$\alpha \int dr e_j(r) F(r) = \sum_i C_{ii}^{-1} [g_i - u_i - c_i] \int dr e_j(r) K_i(r)$$

ó

$$\alpha u_j = \sum_i S_{ij} C_{ii}^{-1} [g_i - u_i - c_i]$$

donde

$$S_{ij} = \int \int K_i(r) K_j(r') G(r, r') dr dr' \quad (30)$$

En forma matricial, tenemos

$$\alpha u = SC_{yy}^{-1} [g - u - c] \quad (31)$$

Para despejar u , obtenemos

$$[\alpha I + SC_{yy}^{-1}] u = SC_{yy}^{-1} [g - c]$$

$$u = (\alpha I + SC_{yy}^{-1})^{-1} SC_{yy}^{-1} [g - c]$$

sustituyendo en la ecuación (29) resulta que

$$F(r) = \frac{1}{\alpha} K^T(r) C_{yy}^{-1} [g - c - (SC_{yy}^{-1} + \alpha I)^{-1} SC_{yy}^{-1} (g - c)]$$

ó

$$F(r) = \frac{1}{\alpha} K^T(r) C_{yy}^{-1} [I - (SC_{yy}^{-1} + \alpha I)^{-1} SC_{yy}^{-1}] (g - c)$$

Utilizando la identidad $I = (SC_{yy}^{-1} + \alpha I)^{-1} (SC_{yy}^{-1} + \alpha I)$, la expresión anterior se reescribe

$$F(r) = K^T(r) C_{yy}^{-1} (SC_{yy}^{-1} + \alpha I)^{-1} (g - c) \quad (32)$$

Sustituyendo esta expresión con argumento r' en la ecuación (22), obtenemos una expresión explícita para la solución regularizada de la función incógnita $m(r)$.

$$\hat{m}(r) = \int dr' K^T(r') C_{yy}^{-1} (S C_{yy}^{-1} + \alpha I)^{-1} (g - c) G(r', r) + N(r)$$

y por la definición (24) tenemos

$$\hat{m}(r) = e^T(r) C_{yy}^{-1} (S C_{yy}^{-1} + \alpha I)^{-1} (g - c) + N(r) \quad (33)$$

Analicemos el comportamiento de $\hat{m}(r)$ cuando $|r| \rightarrow \infty$. El primer término del lado derecho de (33) crece como $\ln|r|$ en el infinito; esto es porque $r \gg r'$, lo que conduce a que

$$\lim_{r \rightarrow \infty} e_i(r) = -(2\pi)^{-1} \ln|r| d_i$$

donde

$$d_i = \int dr' K_i(r') \quad (34)$$

El segundo término del lado derecho de (33) puede ser una constante o puede variar como un polinomio de x y y (coordenadas de r) cuando $|r| \rightarrow \infty$. Por lo tanto, para que $\hat{m}(r)$ sea constante en el infinito, primero la función $N(r)$ debe ser constante, que denotamos por ξ , por lo cual tenemos

$$c_i = \xi d_i \quad (35)$$

y el coeficiente de $\ln|r|^{-1}$ del primer término de la ecuación (33) debe ser igual a cero

$$d^T C_{yy}^{-1} (S C_{yy}^{-1} + \alpha I)^{-1} (g - \xi d) = 0$$

o, de otra forma,

$$d^T(S + \alpha C_{yy})^{-1}(g - \xi d) = 0 \quad (36)$$

De esta ecuación, despejamos la constante ξ

$$\xi = \frac{d^T(S + \alpha C_{yy})^{-1}g}{d^T(S + \alpha C_{yy})^{-1}d}$$

y la sustituimos en (33)

$$\hat{m}(r) = e^T(r)(S + \alpha C_{yy})^{-1} \left[g - \frac{d^T(S + \alpha C_{yy})^{-1}g}{d^T(S + \alpha C_{yy})^{-1}d} d \right] + \frac{d^T(S + \alpha C_{yy})^{-1}g}{d^T(S + \alpha C_{yy})^{-1}d}$$

Factorizando algunos términos y reordenando, la expresión final para la solución regularizada de la función incógnita $m(r)$ es

$$\hat{m}(r) = e^T(r)(S + \alpha C_{yy})^{-1}g + \frac{1 - e^T(r)(S + \alpha C_{yy})^{-1}d}{d^T(S + \alpha C_{yy})^{-1}d} d^T(S + \alpha C_{yy})^{-1}g \quad (37)$$

I.4 Objetivo de la Tesis

El objetivo del trabajo de esta tesis es la implementación computacional del método de inversión (estimador óptimo) desarrollado en la sección I.3. Para ésto, se hace un análisis de la expresión (37) y se desarrolla un procedimiento computacional con el que se resuelven dos problemas inversos, uno sencillo y otro más complicado para determinar su bondad, en implementación y aplicación, respectivamente.

Al mismo tiempo, se busca optimizar el código para utilizar eficientemente los recursos computacionales (CPU y memoria) disponibles.

Dadas las numerosas integrales múltiples a calcular (ver expresiones 24, 30, 34 y 37) así como productos puntos, la optimización del procedimiento a desarrollar es punto esencial en este trabajo de tesis.

Capítulo II

Procedimiento de Inversión.

Validación del Procedimiento

Numérico Computacional

El problema a tratar es una ecuación de Fredholm de primera clase

$$\int K_i(r) m(r) dr = g_i \quad (38)$$

que es continua en los argumentos de la función incógnita $m(r)$ y discreta en los datos g_i , donde $r = (x, y)$, $dr = dx dy$ y el índice i corresponde al punto (x_i, y_i) donde se hace la observación g_i .

En esta tesis llamamos datos u observaciones a los elementos del vector g , aunque éstos sean obtenidos analítica o numéricamente, es decir, no necesariamente de la experiencia.

El *problema directo* consiste en calcular g_i dados el kernel $K_i(r)$ y la función $m(r)$. El *problema inverso* consiste en estimar la función $m(r)$ dados el kernel $K_i(r)$ y los valores g_i .

En el presente capítulo, se describe la implementación del procedimiento que se desarrolla a partir de la expresión (37) obtenida en el capítulo anterior aplicándolo a un problema inverso cuya solución es conocida analíticamente.

Para validar el procedimiento, se define un problema a través de un kernel y una función incógnita, se resuelve el problema directo y, utilizando los resultados del capítulo anterior, se resuelve el problema inverso. Finalmente se compara la solución del problema inverso con la introducida originalmente, ya sea por inspección visual de la solución o del error de la solución. Adicionalmente, el análisis de residuales juega un papel evaluador importante, particularmente si las observaciones tienen ruido o error.

II.1 Definición del Problema

El problema seleccionado para la validación numérica tiene las siguientes características. El kernel es una función gaussiana bivalente

$$K_i(x, y) = e^{-l_x(x-x_i)^2} e^{-l_y(y-y_i)^2} \quad (39)$$

donde las desviaciones estándares implícitas (l_x, l_y) determinan el ancho efectivo del kernel

$$l_x = \frac{1.0}{2\sigma_x^2} \quad l_y = \frac{1.0}{2\sigma_y^2} \quad (40)$$

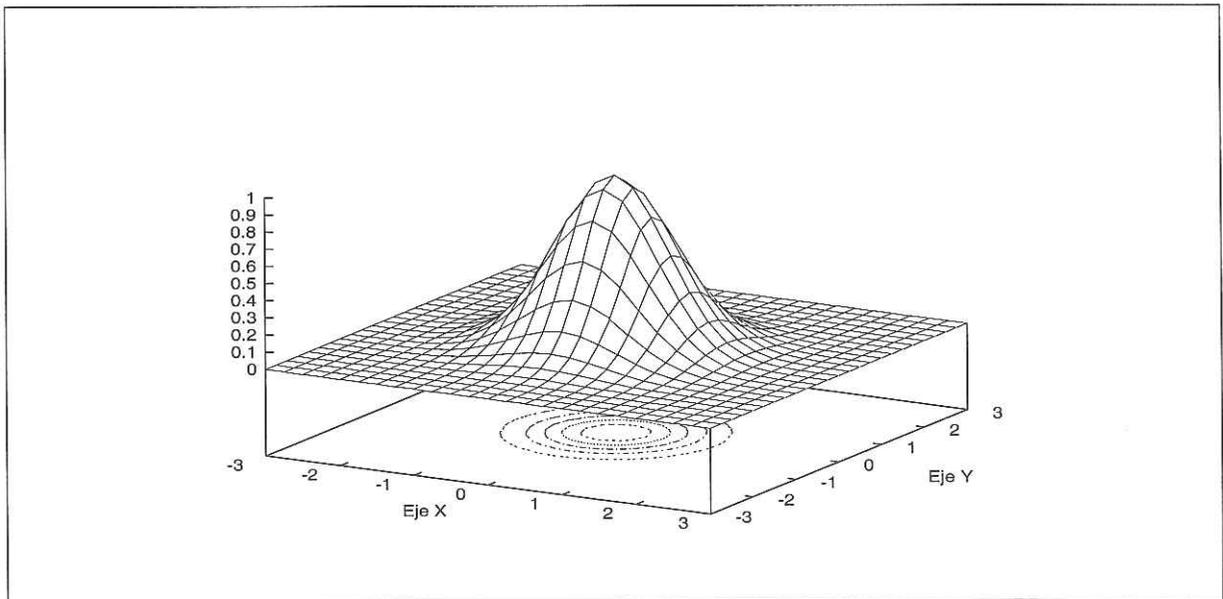


Figura 1: Gráfica de una Función Gaussiana bivalente: $x_i = 0$, $y_i = 0$, $l_x = 1.0$ y $l_y = 1.0$

Este kernel tiene su valor máximo en el punto (x_i, y_i) .

Como función incógnita $m(x, y)$ se elige la función no-lineal

$$m(x, y) = a + b x + c y + d x y \quad (41)$$

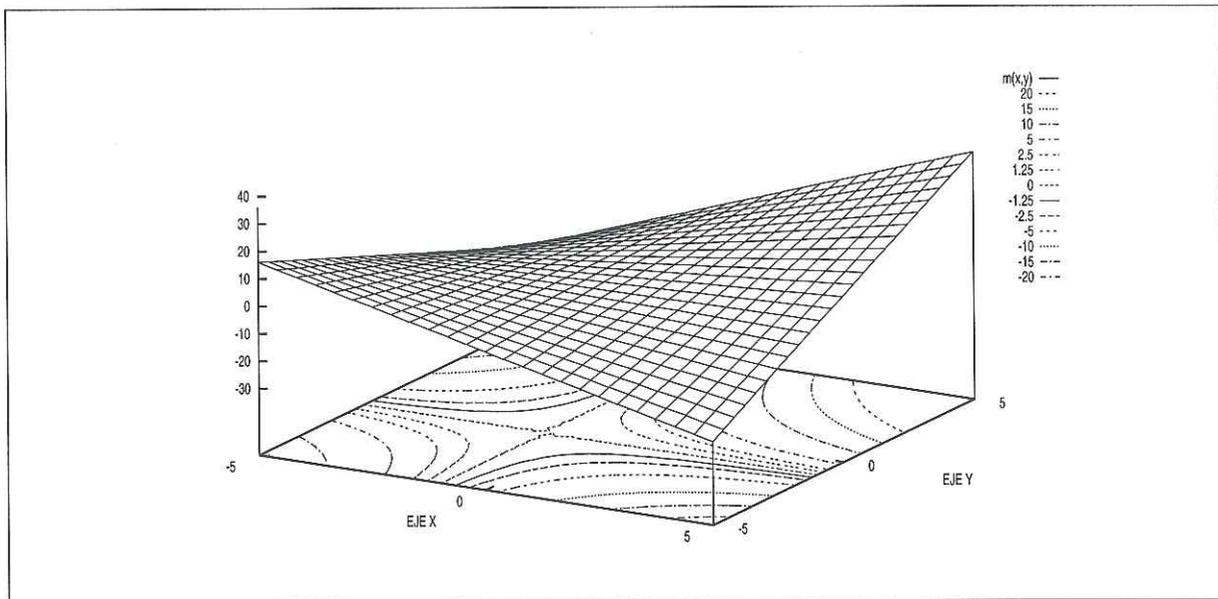


Figura 2: Gráfica de la función $m(x, y) = a + bx + cy + dxy$, con parámetros $a = b = c = d = 1.0$ en la región de interés

Las razones para escoger el kernel (39) y la función incógnita (41) son:

- El kernel tiene la forma de "campana"; esta forma de kernel se presenta comúnmente en problemas físicos.
- La integración sobre todo el espacio del producto $K_i(r)m(r)$ tiene solución analítica. Esto nos permite construir el vector de observaciones g sin aproximaciones numéricas.
- La función $m(r)$ puede representar diferentes superficies variando los parámetros a, b, c y d .
- No existen en la literatura funciones de prueba que se hayan utilizado extensivamente para propósitos de validación en el caso bivariante, al contrario de lo

que sucede en el caso unidimensional. Por ello, debimos formar una función de prueba con las tres características anteriores.

Este problema inverso se puede complicar si se realiza una rotación de ejes y, luego, se plantea su solución numérica, cosa que no se hizo en la tesis.

II.2 Solución Analítica del Problema Directo

El problema directo consiste en determinar g_i dados $K_i(r)$ y $m(r)$

$$g_i = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-l_x(x-x_i)^2} e^{-l_y(y-y_i)^2} (a + bx + cy + dxy) dx dy \quad (42)$$

donde $g_i = g(x_i, y_i)$. Esta doble integral es de fácil solución ya que se conoce el cálculo del promedio de una función gaussiana. El término no-lineal se desacopla en integrales del mismo tipo que las correspondientes a los términos lineales. El resultado es

$$g_i = a \sqrt{\frac{\pi}{l_x}} \sqrt{\frac{\pi}{l_y}} + b \sqrt{\frac{\pi}{l_x}} \sqrt{\frac{\pi}{l_y}} x_i + c \sqrt{\frac{\pi}{l_x}} \sqrt{\frac{\pi}{l_y}} y_i + d \sqrt{\frac{\pi}{l_x}} \sqrt{\frac{\pi}{l_y}} x_i y_i$$

Factorizando términos la expresión anterior queda

$$g_i = \frac{\pi}{\sqrt{l_x l_y}} [a + bx_i + cy_i + dx_i y_i] \quad (43)$$

Utilizamos los siguientes parámetros:

$$l_x = 1.0 \quad l_y = 1.0$$

$$a = 1.0 \quad b = 1.0 \quad c = 1.0 \quad d = 1.0$$

El kernel es simétrico con respecto al punto (x_i, y_i) , teniendo como desviación estándar

$$\sigma_x = \frac{1.0}{\sqrt{2}} \quad \sigma_y = \frac{1.0}{\sqrt{2}}$$

El plano donde se calcularán las observaciones g_i esta delimitado por

$$|x_S| \leq 5 \quad |y_S| \leq 5$$

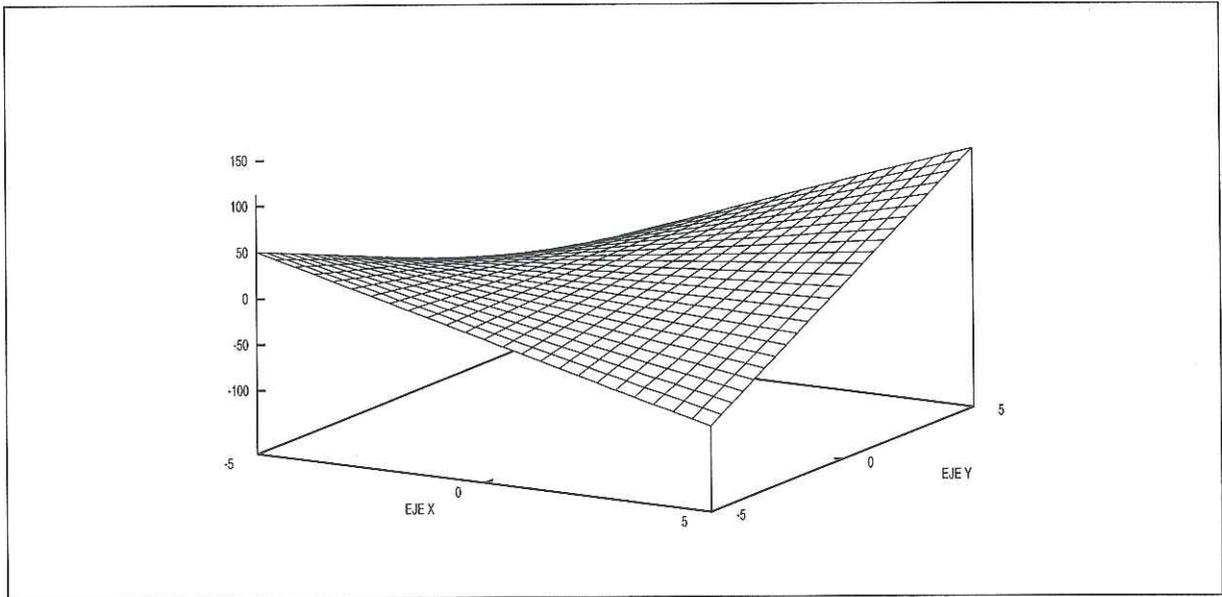


Figura 3: Gráfica de la función g , con parámetros $l_x = l_y = 1.0$ y $a = b = c = d = 1.0$, sobre la región donde se obtienen las observaciones.

II.3 Planteamiento del Problema Inverso

Dado el kernel

$$K_i(x, y) = e^{-(x-x_i)^2} e^{-(y-y_i)^2} \quad (44)$$

y la función observación

$$g_i = \pi [1.0 + x_i + y_i + x_i y_i] \quad (45)$$

obtenemos la estimación $\hat{m}(x, y)$ de la función incógnita mediante la expresión (37)

desarrollada en el capítulo anterior

$$\hat{m}(r) = e^T(r)(S + \alpha C_{yy})^{-1}g + \frac{1 - e^T(r)(S + \alpha C_{yy})^{-1}d}{d^T(S + \alpha C_{yy})^{-1}d} d^T(S + \alpha C_{yy})^{-1}g \quad (37)$$

donde C_{yy} es la matriz de covarianza del error de los datos, g es el vector de datos y α es el parámetro de regularización. La definición de los vectores d , $e(r)$ y la matriz S son como sigue

$$e_i(r) = \int K_i(r')G(r, r') dr' \quad (24)$$

$$S_{ij} = \int \int K_i(r)K_j(r')G(r, r') drdr' \quad (30)$$

$$d_i = \int K_i(r') dr' \quad (34)$$

donde $G(r, r')$ es la función de Green del laplaciano ∇^2 en 2-D

$$G(r, r') = -\frac{1}{2\pi} \ln |r - r'| = G(r', r) \quad (46)$$

Como ya se indicó, en los siguientes desarrollos se considera que las observaciones g_i son obtenidos dentro de la región $|x_i| \leq 5$ y $|y_i| \leq 5$; esta es la misma región donde se desea determinar la función incógnita $m(x, y)$.

Analicemos estas integrales para este caso particular. La expresión d_i corresponde a obtener el factor de normalización de una distribución gaussiana; en este caso particular la solución es

$$d_i = \pi \quad (47)$$

Como se observa, el resultado no depende del índice i .

Las expresiones $e_i(r)$ y S_{ij} no se pueden calcular analíticamente, por lo que deben obtenerse numéricamente.

Un aspecto a considerar al resolver numéricamente las expresiones $e_i(r)$ y S_{ij} son los límites de integración. En la integración numérica no podemos tener límites de integración infinitos; éstos deben truncarse, con algún criterio específico, a un valor finito. Se observa que tanto el integrando de $e_i(r)$ como el de S_{ij} tienen la función gaussiana, la cual decae a cero conforme uno se aleja del punto donde tiene su valor máximo. La rapidez con que decae a cero depende de los valores de l_x y l_y . El parámetro l_x determina la rapidez con que decae a cero en la dirección del eje x y l_y en la dirección del eje y . Valores pequeños de l_x y l_y corresponden a valores grandes de σ_x y σ_y , respectivamente, por lo que la función gaussiana se hace más ancha; al contrario valores grandes de l_x y l_y corresponden a valores pequeños de σ_x y σ_y , respectivamente, por lo que la función gaussiana se hace más angosta. Por lo anterior, valores grandes de l_x y l_y hacen que la función gaussiana decaiga rápidamente a cero.

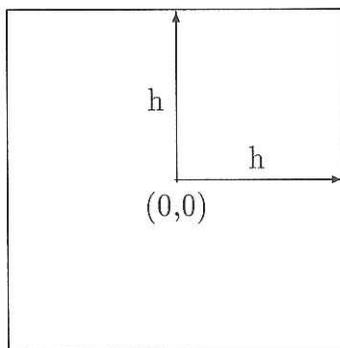
Para la validación, tenemos $l_x = 1.0$ y $l_y = 1.0$, que corresponden, respectivamente, a $\sigma_x = 1/\sqrt{2}$ y $\sigma_y = 1/\sqrt{2}$. Con estos valores, la función gaussiana normalizada tiene un valor del orden de 10^{-8} cuando nos alejamos 5 unidades del punto donde la función gaussiana tiene su valor máximo.

Esto nos da la libertad de integrar en un rango finito, ya que la contribución que hace el integrando a la integral total, en regiones alejadas del punto donde tiene su máximo valor la función gaussiana, es despreciable.

II.4 Métodos de Integración Numérica

Para poder utilizar la expresión (37) en la computadora, debemos obtener numéricamente $e_i(r)$ y S_{ij} . Para ésto, debemos seleccionar un método para integrar numéricamente funciones bivariantes. Las regiones donde se van a calcular las expresiones $e_i(r)$ y S_{ij} se dividen en celdas cuadradas de lado $2h$ a las cuales se les aplican los métodos de integración. Cada celda comprende la región R

$$R : |x| \leq h, |y| \leq h$$



En esta tesis se utilizan dos métodos de integración numérica, los cuales se describen a continuación.

El primer método establece que la doble integral de una función $f(x, y)$ en la región R se aproxima numéricamente por (Abramowitz y Stegun, 1970)

$$\int \int_R f(x, y) \, dx dy = 4h^2 \left[\sum_{i=1}^4 w_i f(x_i, y_i) + O(h^4) \right] \quad (48)$$

donde los valores para las w_i y los puntos (x_i, y_i) son

(x_i, y_i)	w_i
$(h \sqrt{\frac{1}{3}}, h \sqrt{\frac{1}{3}})$	$\frac{1}{4}$
$(h \sqrt{\frac{1}{3}}, -h \sqrt{\frac{1}{3}})$	$\frac{1}{4}$
$(-h \sqrt{\frac{1}{3}}, h \sqrt{\frac{1}{3}})$	$\frac{1}{4}$
$(-h \sqrt{\frac{1}{3}}, -h \sqrt{\frac{1}{3}})$	$\frac{1}{4}$

en otras palabras se evalúa la función a integrar en cuatro puntos dentro de la celda unitaria de integración y se hace una suma ponderada, con factores de ponderación w_i , de estas evaluaciones. El error del método es del orden de h^6 . En adelante, nos referimos a este método como *método de cuatro puntos*.

Se ha elegido la siguiente integral doble

$$I = \int_0^5 \int_0^5 (e^{-x} + e^{-y}) dx dy \quad (49)$$

para una validación de este método de integración. El resultado analítico es

$$I = 10.0 (1.0 - e^{-5}) \quad (50)$$

Se ha elegido esta integral doble por tener un integrando continuo y por tener una solución analítica.

La siguiente tabla muestra, para diferentes valores de h , la diferencia entre el valor exacto y el valor calculado numéricamente, éste es, el error.

$2h$	Error
5.00	7.57370×10^{-1}
2.50	7.45427×10^{-2}
1.00	2.22811×10^{-3}
0.50	1.42568×10^{-4}
0.25	8.96352×10^{-6}
0.10	2.29848×10^{-7}

Como se observa, el error disminuye conforme el valor de h disminuye, mostrando convergencia por el lado positivo para este rango de valores de h .

El segundo método de integración establece que la integral doble de una función $f(x, y)$ en la región R esta dada por (Abramowitz y Stegun, 1970)

$$\int \int_R f(x, y) \, dx dy = 4h^2 \left[\sum_{i=1}^9 w_i f(x_i, y_i) + O(h^4) \right] \quad (51)$$

donde los valores para las w_i y los puntos (x_i, y_i) son

(x_i, y_i)	w_i
$(0, 0)$	$\frac{4}{9}$
$(\pm h, \pm h)$	$\frac{1}{36}$
$(\pm h, 0)$	$\frac{1}{9}$
$(0, \pm h)$	$\frac{1}{9}$

En otras palabras, se evalúa la función a integrar en nueve puntos dentro de la celda de integración y se hace una suma ponderada de estos valores. Al igual que el método de cuatro puntos este método tiene un error del orden de h^6 . En lo siguiente, se hace referencia a este esquema de integración como "*método de nueve puntos*".

Para validar este método usamos de nuevo la expresión (49). La siguiente tabla

muestra, para diferentes valores de h , la diferencia entre el valor exacto y el valor calculado numéricamente

$2h$	Error
5.00	-1.19302×10^0
2.50	-1.13204×10^{-1}
1.00	-3.34881×10^{-3}
0.50	-2.13958×10^{-4}
0.25	-1.34469×10^{-5}
0.10	-3.44780×10^{-7}

De nuevo se observa convergencia (esta vez por el lado negativo) ya que disminuye el error conforme el valor de h disminuye.

Los errores obtenidos con ambos métodos de integración son similares para un mismo valor de h , siendo el método de cuatro puntos el que presenta mayor exactitud.

II.5 Cálculo Numérico de Integrales

En esta sección, se define el procedimiento para la computación de las expresiones $e_i(r)$ y S_{ij} , y se muestran resultados numéricos.

II.5.1 Cálculo de $e_i(r)$

La expresión para $e_i(r)$ está definida en la sección (I.3)

$$e_i(r) = \int K_i(r')G(r, r') dr' \quad (24)$$

para el caso que nos ocupa

$$e_i(x, y) = \left(-\frac{1}{4\pi}\right) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-l_x(x'-x)^2} e^{-l_y(y'-y)^2} \ln[(x'-x)^2 + (y'-y)^2] dx' dy' \quad (52)$$

Como ya se mencionó, esta doble integral no tiene solución analítica conocida por lo que debe ser resuelta numéricamente. El integrando tiene una singularidad no esencial en el punto $(x', y') = (x, y)$, por lo que la integración se realiza en dos partes:

- a) Se contruye una solución analítica aproximada para $e_i(x, y)$ alrededor del punto singular $(x', y') = (x, y)$, aproximando con una serie de Taylor al kernel $K_i(x, y)$ alrededor de dicho punto.
- b) Al resultado anterior, se suma el resultado de la integración numérica del resto de la región.

Para construir la solución analítica alrededor del punto singular, se aproxima el kernel $K_i(r)$ por un paraboloides alrededor del punto singular utilizando un desarrollo de Taylor hasta términos cuadráticos. A la doble integral (52) se le asigna una región de integración cuadrada de lado $2h$, siendo h pequeño, alrededor del punto singular. Llamaremos e_0 a este resultado parcial

$$e_0 = \left(-\frac{1}{4\pi} \right) \int_{y-h}^{y+h} \int_{x-h}^{x+h} e^{-l_x(x'-x_i)} e^{-l_y(y'-y_i)} \ln[(x' - x)^2 + (y' - y)^2] dx' dy' \quad (53)$$

Haciendo el cambio de variables

$$u = x' - x_i \quad v = y' - y_i$$

la expresión para e_0 queda

$$e_0 = \left(-\frac{1}{4\pi} \right) \int_{x-h-x_i}^{x+h-x_i} \int_{y-h-y_i}^{y+h-y_i} e^{-l_x u^2} e^{-l_y v^2} \ln[(u - \beta_1)^2 + (v - \beta_2)^2] dv du$$

donde

$$\beta_1 = (x - x_i) \quad \beta_2 = (y - y_i)$$

Como se observa el integrando tiene un punto singular en el punto (β_1, β_2) , por lo que hacemos un desarrollo de Taylor hasta términos de segundo orden de la función

$$f(u, v) = e^{-l_x u^2} e^{-l_y v^2}$$

alrededor de este punto, obteniendo

$$f(u, v) = e^{-l_x \beta_1^2} e^{-l_y \beta_2^2} \left[1 - 2l_x \beta_1 (u - \beta_1) - 2l_y \beta_2 (v - \beta_2) - (l_x - 2l_x^2 \beta_1^2)(u - \beta_1)^2 \right. \\ \left. - (l_y - 2l_y^2 \beta_2^2)(v - \beta_2)^2 + (4l_x l_y \beta_1 \beta_2)(u - \alpha)(v - \beta) \right]$$

Sustituyendo este desarrollo en la doble integral e_0 , obtenemos una aproximación alrededor del punto (β_1, β_2)

$$e_0 \approx \left(-\frac{1}{4\pi} \right) e^{-l_x \beta_1^2} e^{-l_y \beta_2^2} \int_{x-h-x_i}^{x+h-x_i} \int_{y-h-y_i}^{y+h-y_i} \left[1 - 2l_x \beta_1 (u - \beta_1) \right. \\ \left. - 2l_y \beta_2 (v - \beta_2) - (l_x - 2l_x^2 \beta_1^2)(u - \beta_1)^2 - (l_y - 2l_y^2 \beta_2^2)(v - \beta_2)^2 + \right. \\ \left. (4l_x l_y \beta_1 \beta_2)(u - \beta_1)(v - \beta_2) \right] \ln[(u - \beta_1)^2 + (v - \beta_2)^2] du dv$$

Realizando el cambio de variable

$$s = u - \beta_1 \quad t = v - \beta_2$$

tenemos

$$e_0 \approx \left(-\frac{1}{4\pi} \right) e^{-l_x \beta_1^2} e^{-l_y \beta_2^2} \int_{-h}^h \int_{-h}^h \left[1 - 2l_x \beta_1 s - 2l_y \beta_2 t - (l_x - 2l_x^2 \beta_1^2) s^2 \right. \\ \left. - (l_y - 2l_y^2 \beta_2^2) t^2 + (4l_x l_y \beta_1 \beta_2) s t \right] \ln[s^2 + t^2] ds dt$$

Esta expresión se integra analíticamente, con lo que obtenemos la aproximación

$$e_0 \approx \left(-\frac{1}{4\pi} \right) e^{-l_x \beta_1^2} e^{-l_y \beta_2^2} \left[2h^2 (2 \ln(2h^2) + \pi - 6) + \right. \\ \left. \frac{h^4}{3} \left(\pi + 4 \ln(2h^2) - \frac{20}{3} \right) (2l_x^2 \beta_1^2 + 2l_y^2 \beta_2^2 - l_x - l_y) \right] \quad (54)$$

La expresión (54) nos da el valor aproximado de $e_i(r)$ alrededor del punto singular r .

Para integrar numéricamente sobre el resto de la región, ésta se divide en celdas cuadradas y en cada una de éstas se aplica el método de cuatro puntos, descrito en la sección anterior. Debemos definir los intervalos de integración.

La contribución a la integral total está dada por las regiones que se encuentran alrededor del punto donde la gaussiana tiene su valor máximo y alrededor del punto singular.

Como ya se indicó, se considera que los puntos i , donde se calculan los datos g_i , se encuentran en la región

$$|x_i| \leq 5 \quad |y_i| \leq 5$$

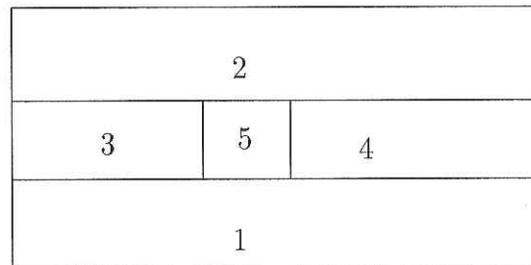
y es en esta misma región donde se quiere determinar $m(r)$. Así, para el cálculo de la expresión $e_i(r)$ tendremos que el punto donde la función gaussiana es máxima y el punto singular, siempre están dentro de la región anterior. Por lo tanto, la contribución a la integral total se da principalmente dentro de esta región y en sus regiones vecinas. Bajo estas consideraciones, se toman los límites de integración para el cálculo de $e_i(r)$ como

$$|x'| \leq 10 \quad |y'| \leq 10$$

Para realizar la integración numérica, se discretiza esta región en una malla cuadrada, donde la distancia nodo-nodo es h . En la implementación computacional, se considera que el punto singular (x, y) corresponde a uno de los nodos de la malla. Esto

nos dice que los puntos donde podamos calcular el valor de la función incógnita $m(x, y)$ están determinados por la distancia nodo-nodo de la malla.

Una vez construida la malla, la región se divide en 5 subregiones, numeradas del 1 al 5. Esquemáticamente:



La integración sobre la subregión 5, cuadrada, de lado $2h$ y que tiene como centro el punto singular $(x', y') = (x, y)$, se resuelve aproximadamente mediante la expresión (54). La integración de las subregiones 1, 2, 3 y 4 se hace por separado utilizando el método de integración de cuatro puntos.

Se resuelve $e_i(r)$ para algunos puntos i y distintos valores de r . En las tablas I, II, III y IV se entregan resultados del cálculo de $e_i(r)$ para algunos puntos en función de h .

Tabla I: $e_i(x, y)$ para $(x_i, y_i) = (0, 0)$ y $(x, y) = (0, 0)$

$2h$	$e_i(x, y)$
2.0	3.237219978889135
1.0	0.142449645928647
0.5	0.142352681605860
0.25	0.144234960100288
0.2	0.144280830528748
0.1	0.144302471969502
0.05	0.144303646274512

Tabla II: $e_i(x, y)$ para $(x_i, y_i) = (0.2, 0.2)$ y $(x, y) = (0, 0)$

$2h$	$e_i(x, y)$
2.0	2.686325741709370
1.0	0.123314185670740
0.5	0.123166250979276
0.25	0.124642337364977
0.2	0.124678373274595
0.1	0.124695637120629
0.05	0.124696662262602

Tabla III: $e_i(x, y)$ para $(x_i, y_i) = (-0.2, -0.2)$ y $(x, y) = (0, 0)$

$2h$	$e_i(x, y)$
2.0	2.686325741709370
1.0	0.123314185670740
0.5	0.123166250979276
0.25	0.124642337364977
0.2	0.124678373274595
0.1	0.124695637120629
0.05	0.124696662262601

Tabla IV: $e_i(x, y)$ para $(x_i, y_i) = (5.0, -3.0)$ y $(x, y) = (1.0, -3.0)$

h	$e_i(x, y)$
2.0	-0.800501223626913
1.0	-0.693179038397092
0.5	-0.693147206317835
0.25	-0.693147182851030
0.2	-0.693147182407961
0.1	-0.693147182223471
0.05	-0.693147182219332

II.5.2 Cálculo de S

Los elementos de la matriz S se definen de la siguiente forma

$$S_{ij} = \int \int K_i(r)K_j(r')G(r, r')dr dr' \quad (30)$$

que es una integral cuádruple. Esta integral se puede escribir en términos de $e_j(r)$

$$S_{ij} = \int K_i(r)e_j(r)dr \quad (55)$$

por lo que para resolver S_{ij} se realiza una integración doble sobre la función $K_i(r)e_j(r)$ donde $e_j(r)$ es, a su vez, una integral doble.

Para obtener S_{ij} se hace la siguiente consideración. Si $e_j(r)$ se discretiza en una malla con una distancia nodo-nodo de h , entonces para integrar S_{ij} se utilizará una malla con una distancia nodo-nodo de $2h$. La integración numérica de (55) se realiza con el método de nueve puntos.

Este esquema de integración nos permite que los nueve puntos muestreados sean nodos de la malla para evaluar $e_j(r)$, lo cual está de acuerdo con la restricción que se hace cuando se resuelve $e_i(r)$ en la sección anterior.

Como ya se mencionó, estamos suponiendo que las observaciones g_i se toman en la región

$$|x_i| \leq 5 \quad |y_i| \leq 5$$

Bajo esta restricción y considerando que la función gaussiana decae rápidamente a cero, las regiones de integración para calcular S_{ij} se consideran de la siguiente forma.

Los intervalos de integración para $e_j(r)$ son

$$-10 \leq x' \leq 10 \quad -10 \leq y' \leq 10$$

y los intervalos de integración para la integración de la función $K_i(r)e_j(r)$ se definen como

$$-9 \leq x \leq 9 \quad -9 \leq y \leq 9$$

esto se hace para evitar que el punto singular esté sobre la frontera de integración de $e_i(r)$

Con todo lo anterior ya definido, se hicieron algunas pruebas en el cálculo de la matriz S para dos valores de h . Los resultados se muestran a continuación.

Para las pruebas numéricas se utilizaron los siguientes puntos i , seleccionados arbitrariamente

i	(x_i, y_i)
1	(-3.0, 5.0)
2	(0.5, 5.0)
3	(-1.0, 2.5)
4	(3.0, 2.5)
5	(-4.0, -1.5)

Para un valor de $h = 0.5$ se obtuvo la siguiente matriz S :

$$S = \begin{pmatrix} -0.093660411 & -1.967978296 & -1.828504432 & -2.940017127 & -2.958389801 \\ -1.968147699 & -0.091866167 & -1.683030503 & -1.983931900 & -3.247749223 \\ -1.828647687 & -1.683019524 & -0.091866192 & -2.177580948 & -2.528104836 \\ -2.940211241 & -1.983931934 & -2.177580948 & -0.091866192 & -3.278551979 \\ -2.958604625 & -3.247755164 & -2.528104836 & -3.278551979 & -0.091866192 \end{pmatrix}$$

Para un valor de $h = 0.25$ se obtuvo la siguiente matriz S :

$$S = \begin{pmatrix} -0.091086567 & -1.968080844 & -1.828619555 & -2.940219956 & -2.958592678 \\ -1.968080844 & -0.091086567 & -1.683000745 & -1.983915615 & -3.247752188 \\ -1.828619580 & -1.683000769 & -0.091086585 & -2.177615793 & -2.528099385 \\ -2.940219984 & -1.983915640 & -2.177615793 & -0.091086585 & -3.278556095 \\ -2.958592709 & -3.247752219 & -2.528099385 & -3.278556095 & -0.091086585 \end{pmatrix}$$

II.6 Generación del Vector de Datos

En este capítulo, los datos g_i se obtienen a partir de la ecuación (45)

$$g(x_i, y_i) = \pi(1.0 + x_i + y_i + x_i y_i)$$

y se van a considerar dos situaciones:

1. Generación de datos sin ruido.
2. Generación de datos contaminados con ruido aleatorio y distribución gaussiana.

El primer paso es determinar en qué puntos (x_i, y_i) del plano se ubican las observaciones g_i . Esto se logra implementando una rutina computacional que determina estos puntos aleatoriamente. Se utiliza el criterio de seleccionar los puntos aleatoriamente para simular una situación realista donde las observaciones no necesariamente pueden ser obtenidas en arreglos regulares sobre el plano.

Los puntos se seleccionan aleatoriamente, utilizando un generador de números aleatorios con distribución homogénea, sobre la región

$$|x| \leq 5 \quad |y| \leq 5$$

la cual se discretiza en una malla, cuyos nodos están espaciados 0.5. El pseudocódigo para determinar estos puntos es

```
Numero de Puntos, N
```

```
Mientras i <= N
```

```
x_i = rand()
```

```
y_i = rand()
```

```
Si ( |x_i| <= 5 y |y_i| <= 5 )
```

```
    Aceptar Punto
```

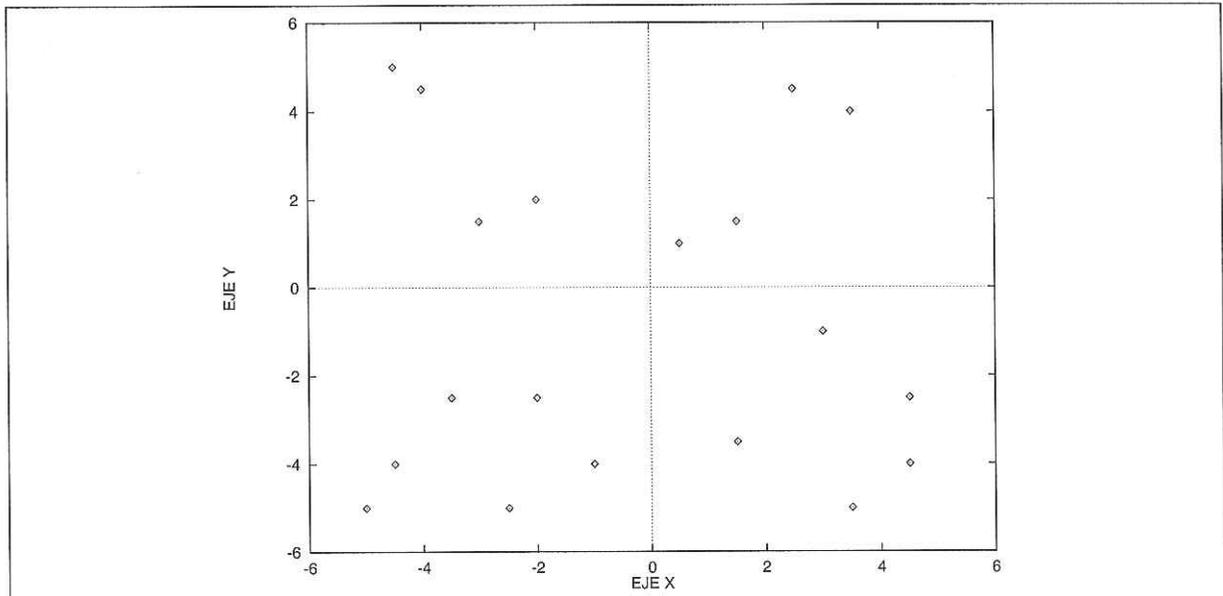
```
Fin Mientras
```

```
Salir
```

Se implementó el pseudocódigo y los puntos (x_i, y_i) obtenidos sobre la región de interés se muestran en la tabla V. La gráfica de estos puntos sobre el plano se muestra en la figura 4.

Tabla V: Puntos de observación para el problema de validación

i	(x_i, y_i)
1	(0.5,1.0)
2	(-4.5,-4.0)
3	(-4.5,5.0)
4	(-1.0,-4.0)
5	(3.5,-5.0)
6	(4.5,-4.0)
7	(-2.5,-5.0)
8	(3.5,4.0)
9	(-2.0,2.0)
10	(-4.0,4.5)
11	(2.5,4.5)
12	(-3.5,-2.5)
13	(1.5,1.5)
14	(-2.0,-2.5)
15	(3.0,-1.0)
16	(1.5,-3.5)
17	(-5.0,-5.0)
18	(-3.0,1.5)
19	(4.5,-2.5)

Figura 4: Gráfica de los puntos (x_i, y_i) utilizados en el problema de validación

Sobre los puntos anteriores se hace el cálculo de las observaciones. En esta primera etapa, no se les agrega ruido. Evaluando la ecuación (45) en los puntos (x_i, y_i) , las observaciones g_i que se obtienen se muestran en la tabla VI.

Tabla VI: Observaciones sin ruido utilizadas en el problema de validación

i	(x_i, y_i)	g_i
1	(0.5,1.0)	9.4247779607694
2	(-4.5,-4.0)	32.9867228626928
3	(-4.5,5.0)	-65.9734457253856
4	(-1.0,-4.0)	0.0000000000000
5	(3.5,-5.0)	-56.5486677646163
6	(4.5,-4.0)	-51.8362787842316
7	(-2.5,-5.0)	18.8495559215388
8	(3.5,4.0)	70.6858347057703
9	(-2.0,2.0)	-9.4247779607694
10	(-4.0,4.5)	-51.8362787842316
11	(2.5,4.5)	60.4756585816035
12	(-3.5,-2.5)	11.7809724509617
13	(1.5,1.5)	19.6349540849362
14	(-2.0,-2.5)	4.7123889803847
15	(3.0,-1.0)	0.0000000000000
16	(1.5,-3.5)	-19.6349540849362
17	(-5.0,-5.0)	50.2654824574367
18	(-3.0,1.5)	-15.7079632679490
19	(4.5,-2.5)	-25.9181393921158

El siguiente paso es agregar a las observaciones diferentes niveles de ruido. Para la validación se consideran tres niveles de ruido relativo: 0.2 %, 1.0 % y 5.0 %.

Para agregar ruido a las observaciones g_i se utiliza la siguiente expresión

$$r = e \left[(p)(|g_i|) + (0.005)\left(\frac{\gamma}{2.0}\right) \right] \quad (56)$$

donde

e : número aleatorio con distribución gaussiana con media cero y desviación estándar uno.

p : valor que indica el porcentaje de ruido (0.002, 0.01, 0.05).

g_i : valor del dato en (x_i, y_i) .

γ : es el valor absoluto del valor máximo menos el valor mínimo de g en la región de interés. En este problema de validación, se tiene $\gamma = 150.0$.

Los valores constantes utilizados en la ecuación (56) se seleccionaron de tal manera que los errores relativos fuera de magnitud similar al valor de p .

Con el primer término de la ecuación (56), se modela el error instrumental que comúnmente es proporcional a la magnitud de la observación, mientras que el segundo término introduce el error debido al modelado, evitando además que el error sea nulo cuando la magnitud de la observación sea cero.

Los resultados para los diferentes niveles de ruido se muestran en la tabla VII.

Tabla VII: Observaciones con ruido para el problema de validación

i	(x_i, y_i)	g_i 0.2%	g_i 1.0%	g_i 5.0%
1	(0.5,1.0)	9.1773	9.1048	8.4244
2	(-4.5,-4.0)	32.7961	31.6157	31.4774
3	(-4.5,5.0)	-66.1742	-63.1984	-63.8451
4	(-1.0,-4.0)	0.0888	-0.1252	0.7161
5	(3.5,-5.0)	-56.4539	-56.3055	-52.5639
6	(4.5,-4.0)	-52.1909	-53.1168	-49.3323
7	(-2.5,-5.0)	18.7451	19.0795	20.5291
8	(3.5,4.0)	70.1006	70.8908	67.5035
9	(-2.0,2.0)	-8.8850	-8.9748	-9.2678
10	(-4.0,4.5)	-52.1006	-51.7373	-49.3029
11	(2.5,4.5)	60.3367	61.1309	56.5882
12	(-3.5,-2.5)	11.4809	11.2548	11.5372
13	(1.5,1.5)	19.2129	18.9275	20.2864
14	(-2.0,-2.5)	4.8751	4.6261	4.2211
15	(3.0,-1.0)	0.0212	0.4215	-0.5049
16	(1.5,-3.5)	-19.5478	-19.2907	-21.4080
17	(-5.0,-5.0)	50.3390	51.7436	49.3630
18	(-3.0,1.5)	-15.6083	-16.1038	-15.1816
19	(4.5,-2.5)	-25.8554	-25.8577	-24.9287

II.7 Solución del Problema Inverso

Para el caso de datos sin ruido, la estimación $\hat{m}(r)$, el error de la solución y los residuales con $\alpha = 1.0$ se muestran en las figuras 5, 6 y 7.

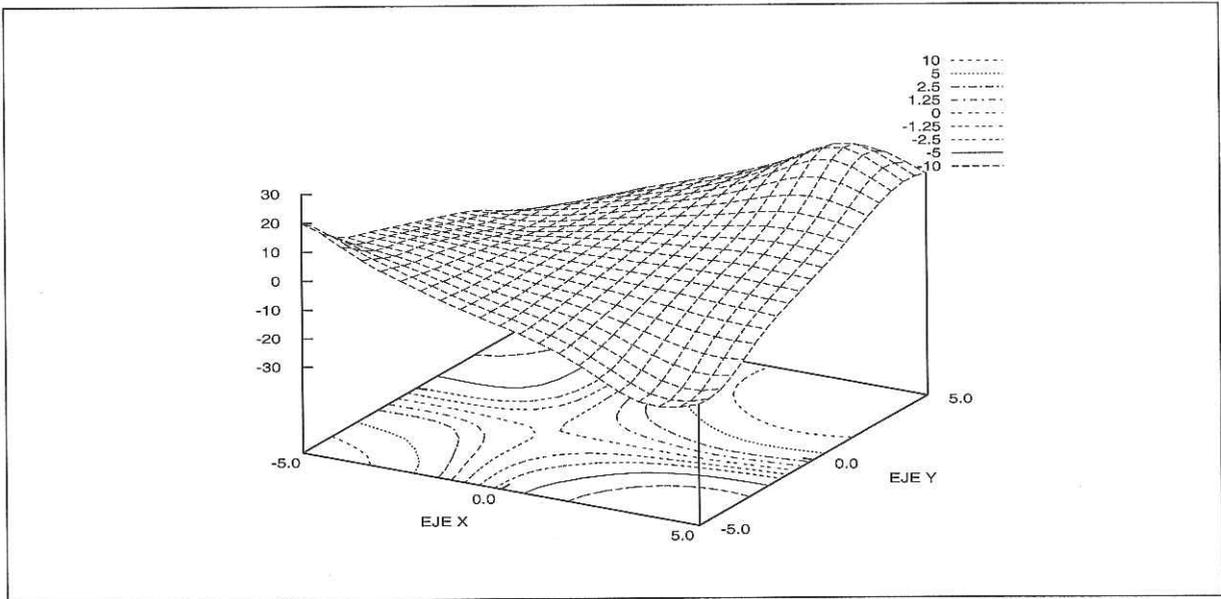


Figura 5: Gráfica de $\hat{m}(r)$ para datos sin ruido; $\alpha = 1.0$

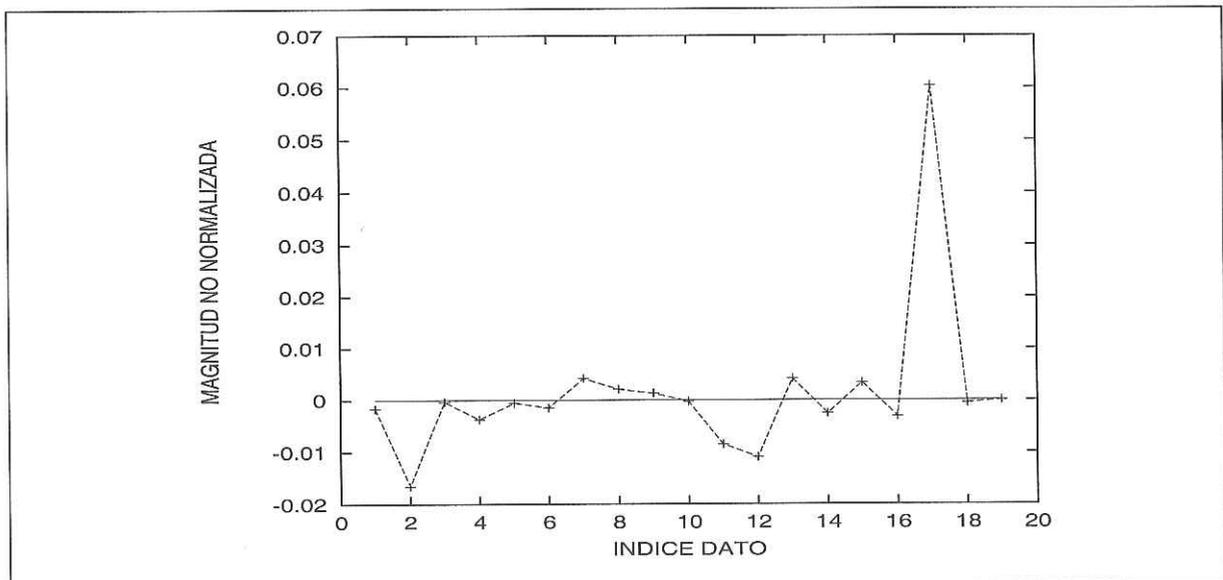


Figura 6: Gráfica de residuales para datos sin ruido; $\alpha = 1.0$

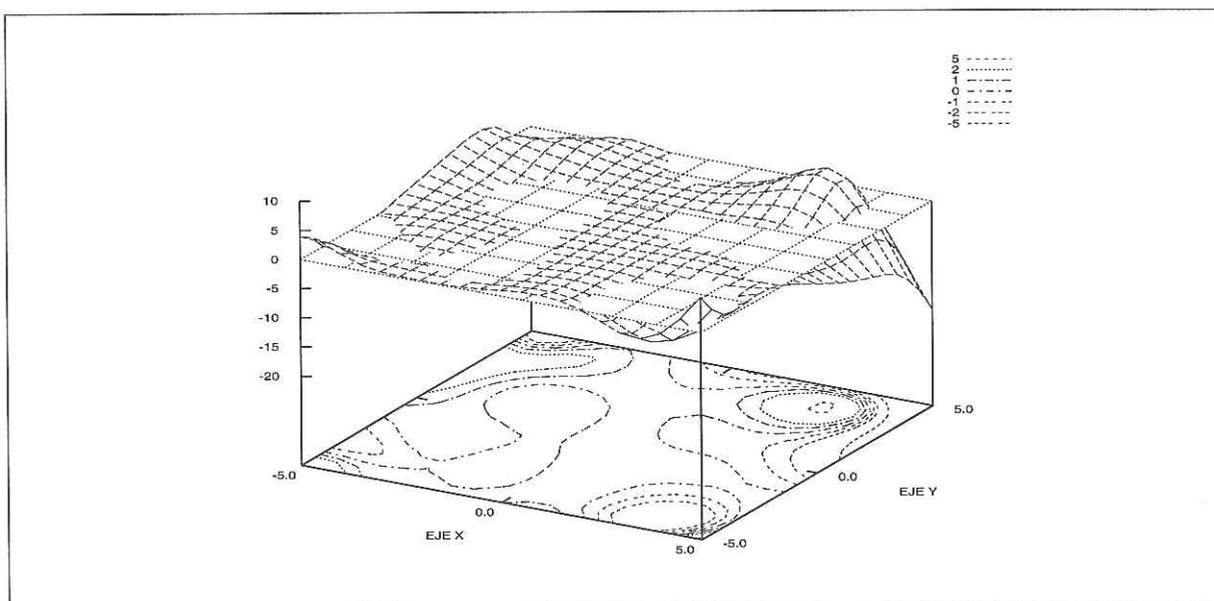


Figura 7: Gráfica del error para datos sin ruido

Para el caso de datos con ruido, se obtiene una estimación $\hat{m}(r)$, el error y su correspondiente gráfica de residuales normalizados, para diferentes valores de α .

Los resultados para el 5.0% de error se entregan en las figuras 8 a la 28.

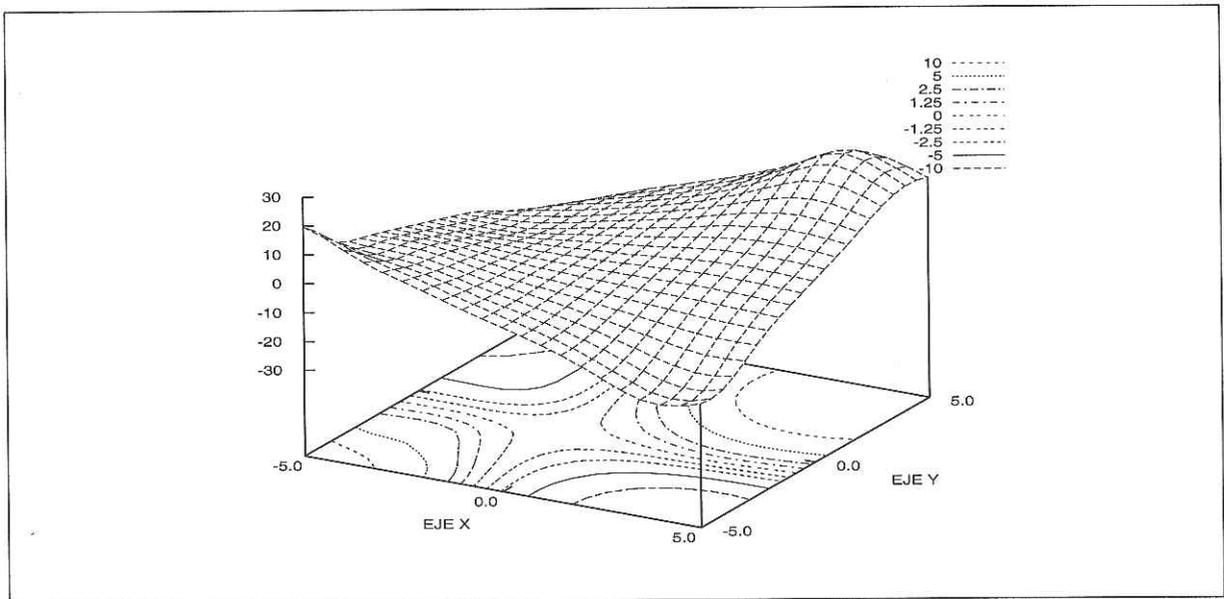


Figura 8: Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 5.0% de error

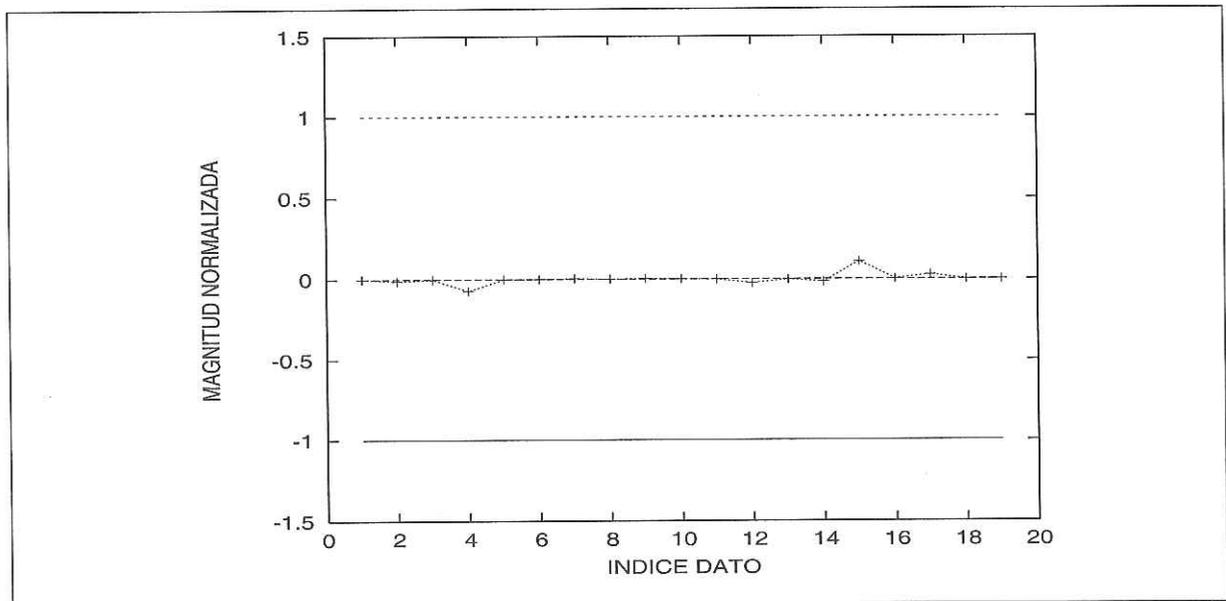


Figura 9: Gráfica de residuales para $\alpha = 0.00001$; 5.0% de error

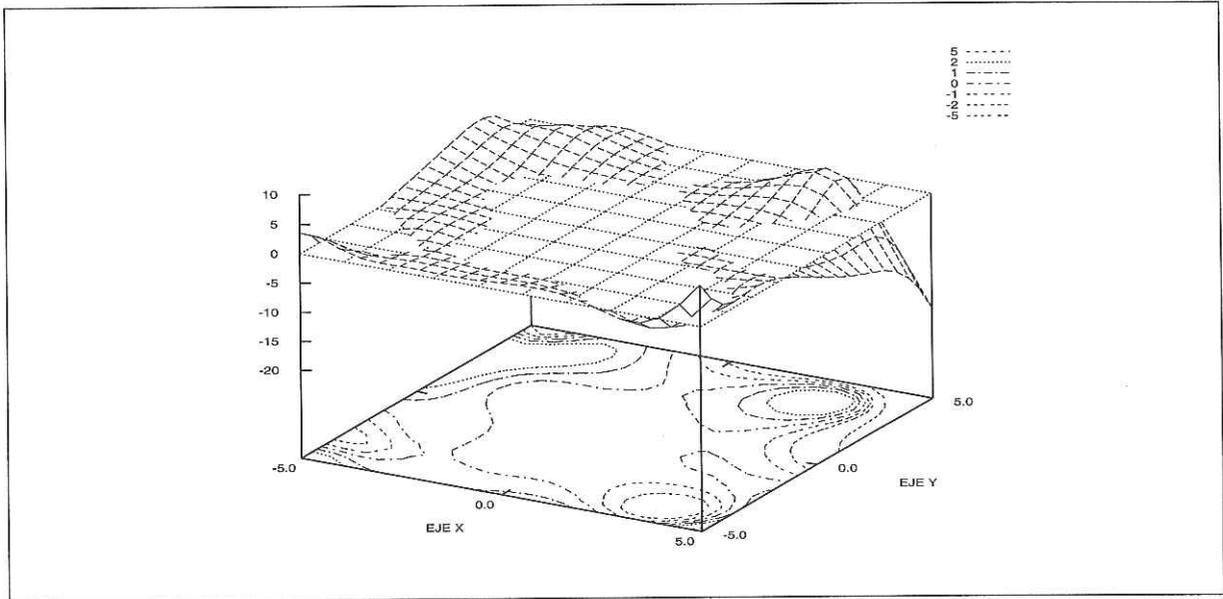


Figura 10: Gráfica del error para $\alpha = 0.00001$; 5.0% de error

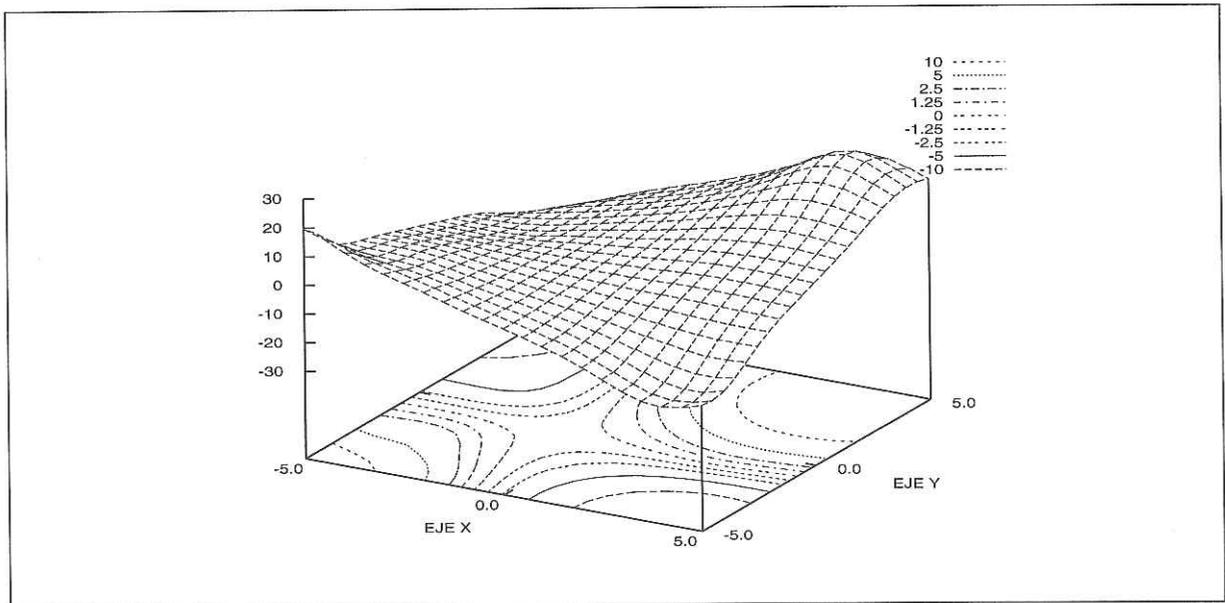


Figura 11: Gráfica de $\hat{m}(r)$ para $\alpha = 0.001$; 5.0% de error

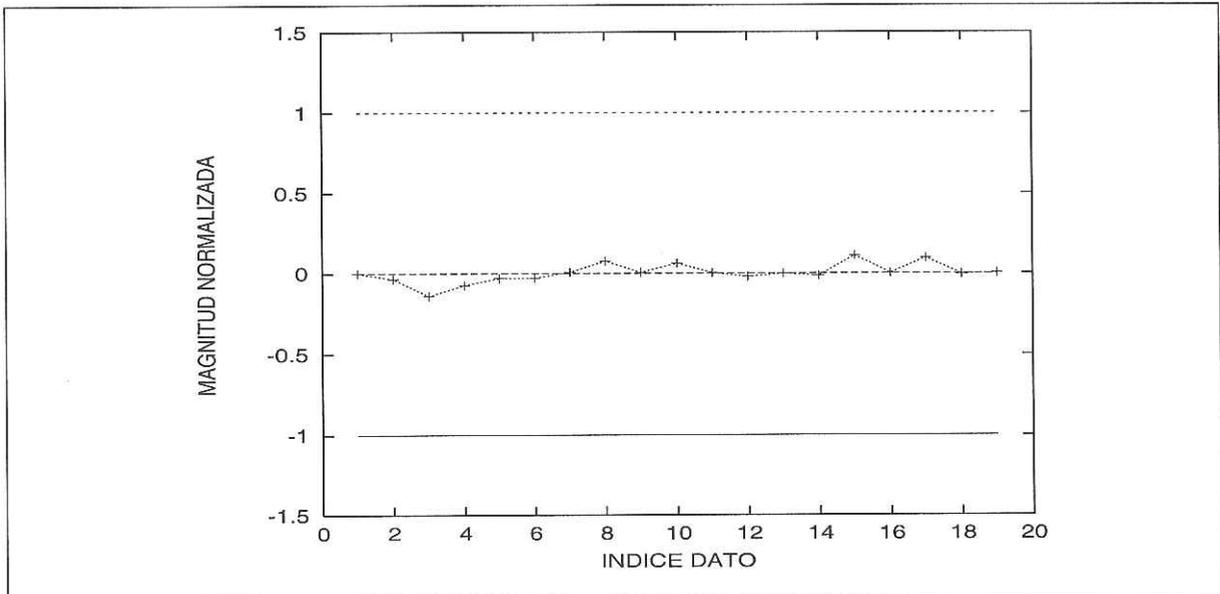


Figura 12: Gráfica de residuales para $\alpha = 0.001$; 5.0% de error

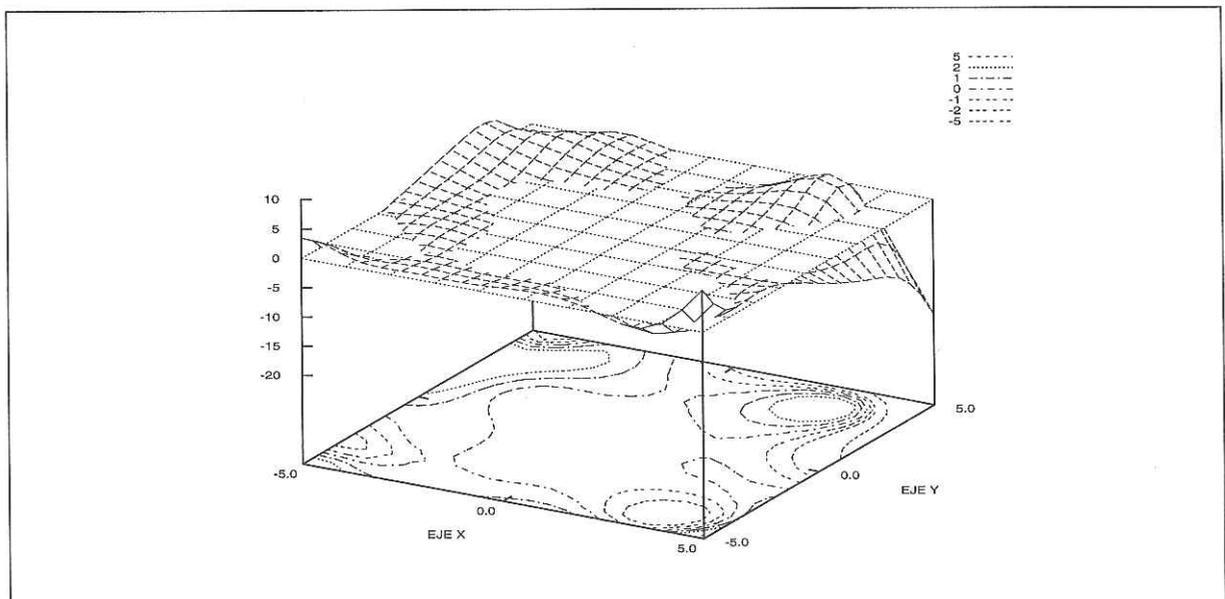


Figura 13: Gráfica del error para $\alpha = 0.001$; 5.0% de error

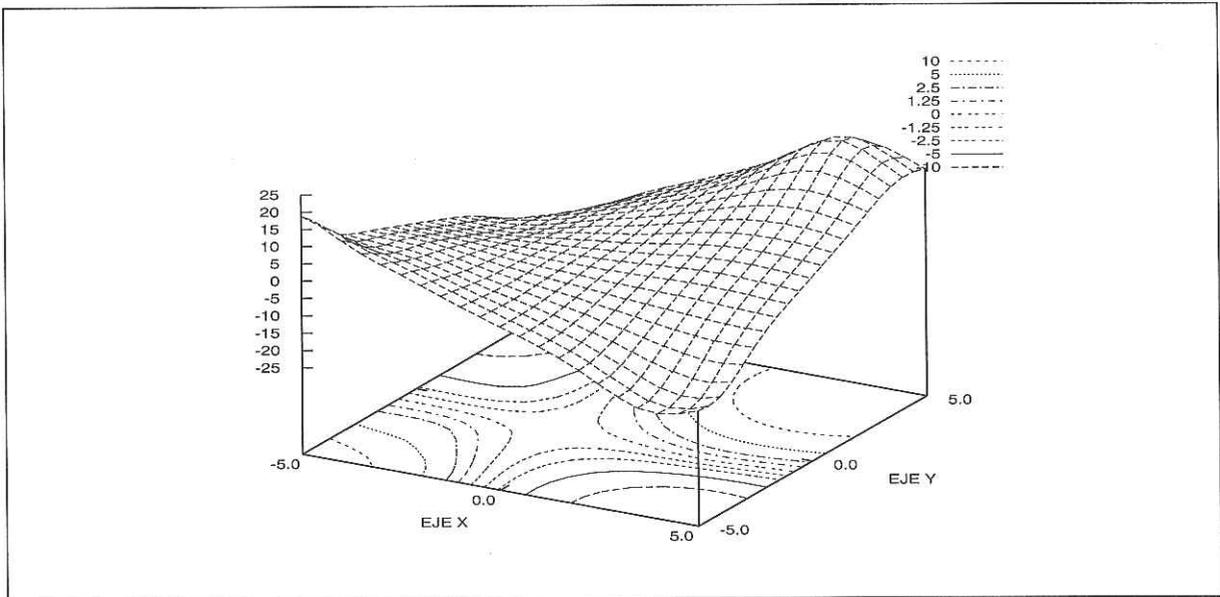


Figura 14: Gráfica de $\hat{m}(r)$ para $\alpha = 0.01$; 5.0% de error

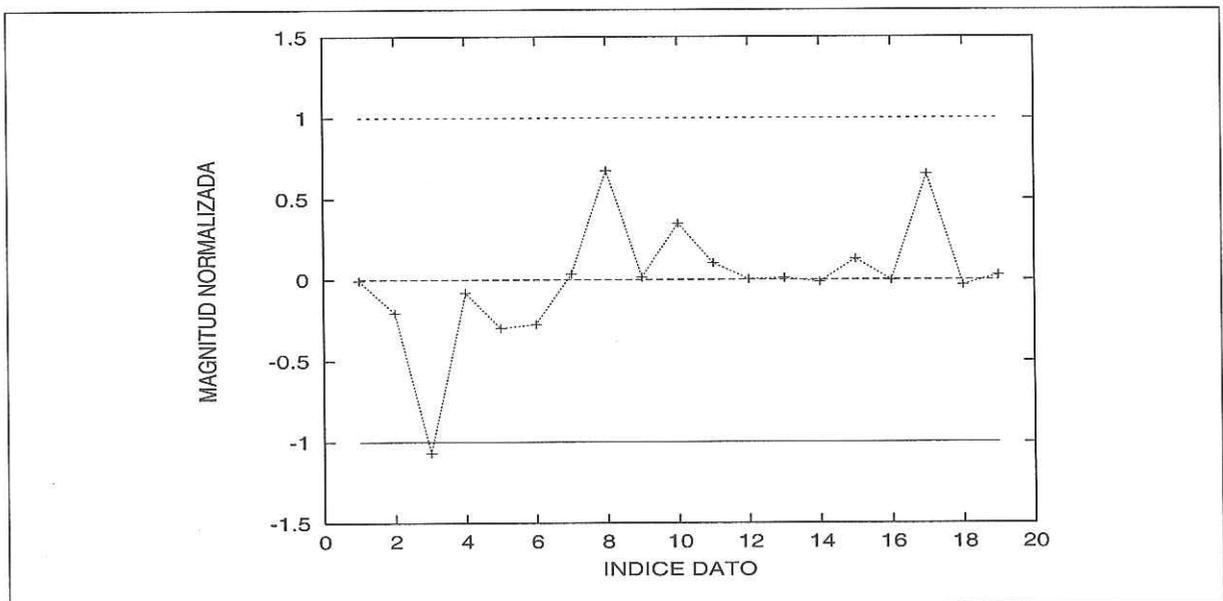


Figura 15: Gráfica de residuales para $\alpha = 0.01$; 5.0% de error

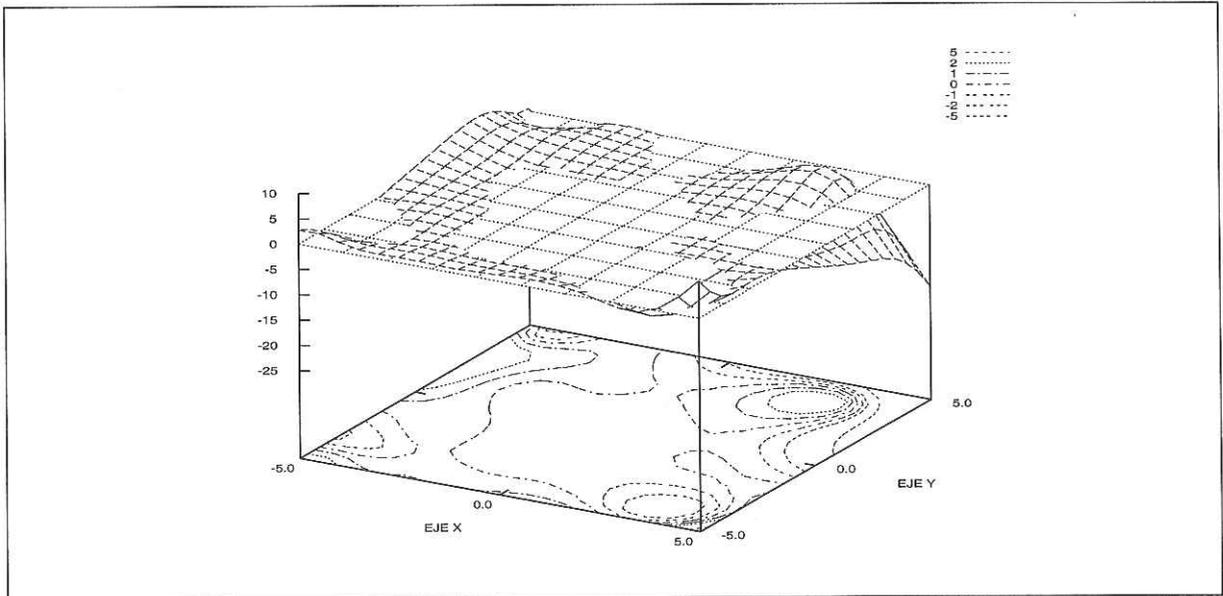


Figura 16: Gráfica del error para $\alpha = 0.01$; 5.0% de error

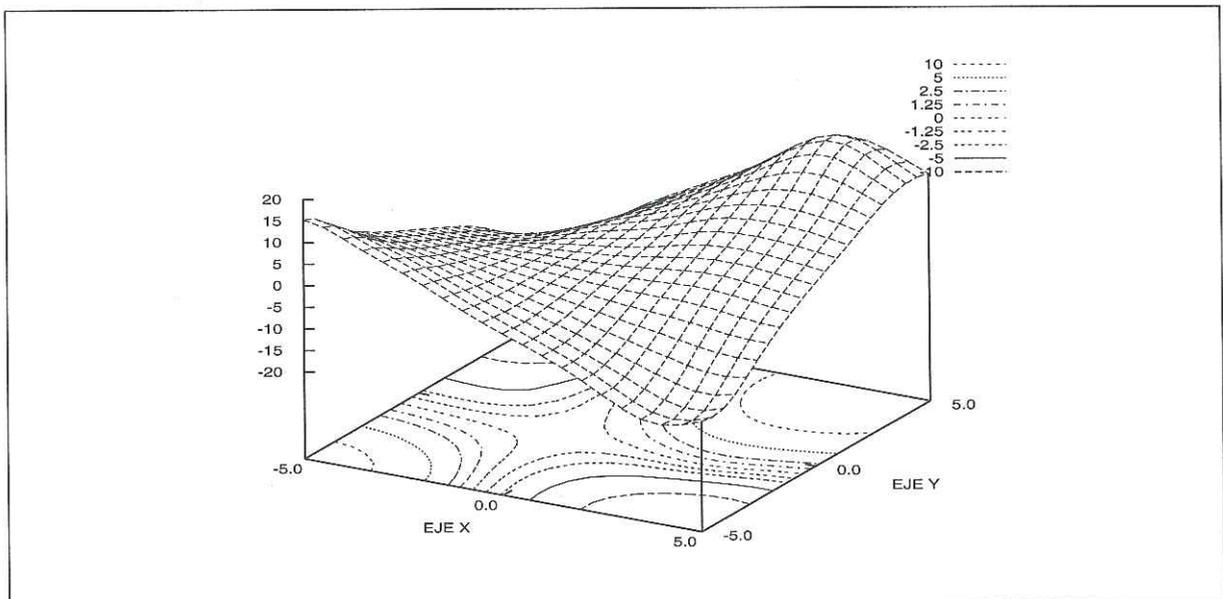


Figura 17: Gráfica de $\hat{m}(r)$ para $\alpha = 0.1$; 5.0% de error

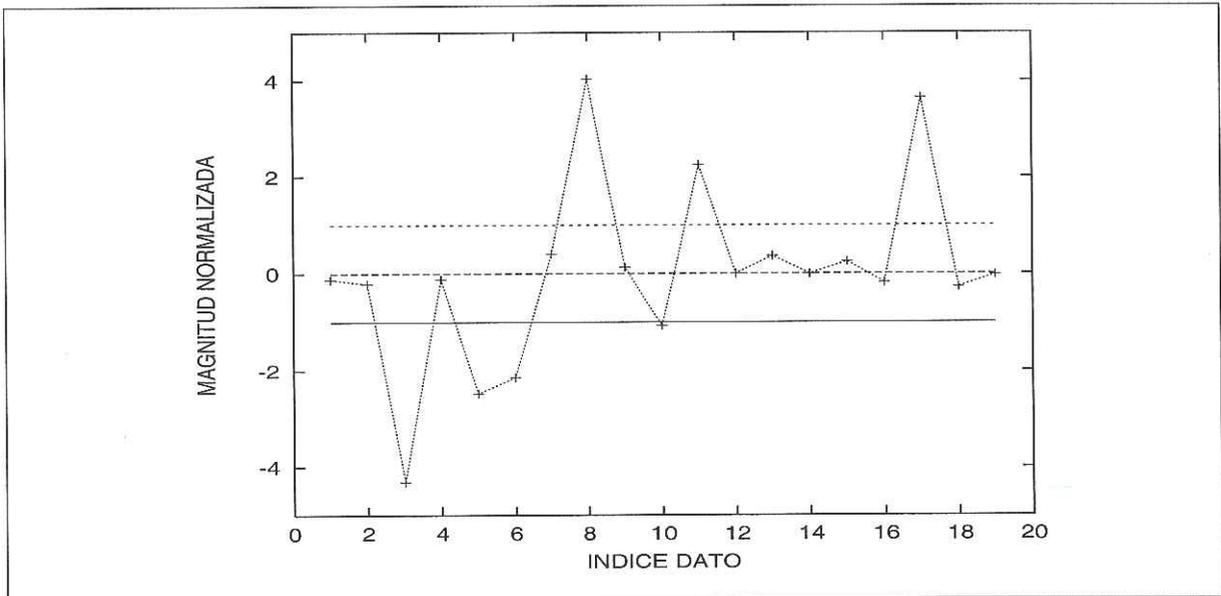


Figura 18: Gráfica de residuos para $\alpha = 0.1$; 5.0% de error

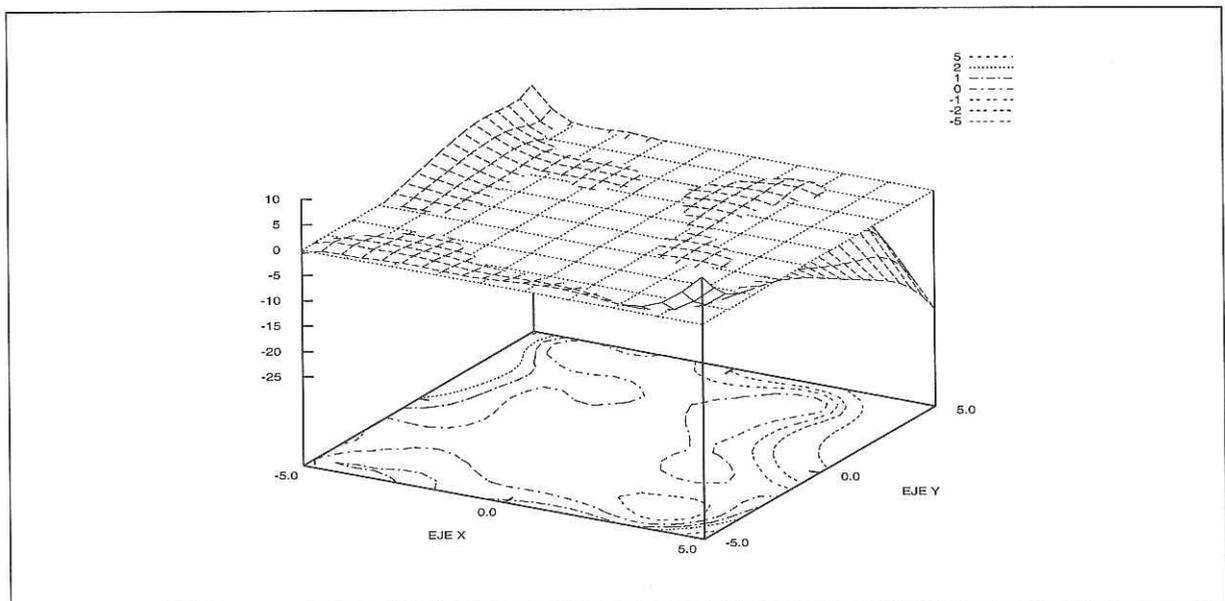


Figura 19: Gráfica del error para $\alpha = 0.1$; 5.0% de error

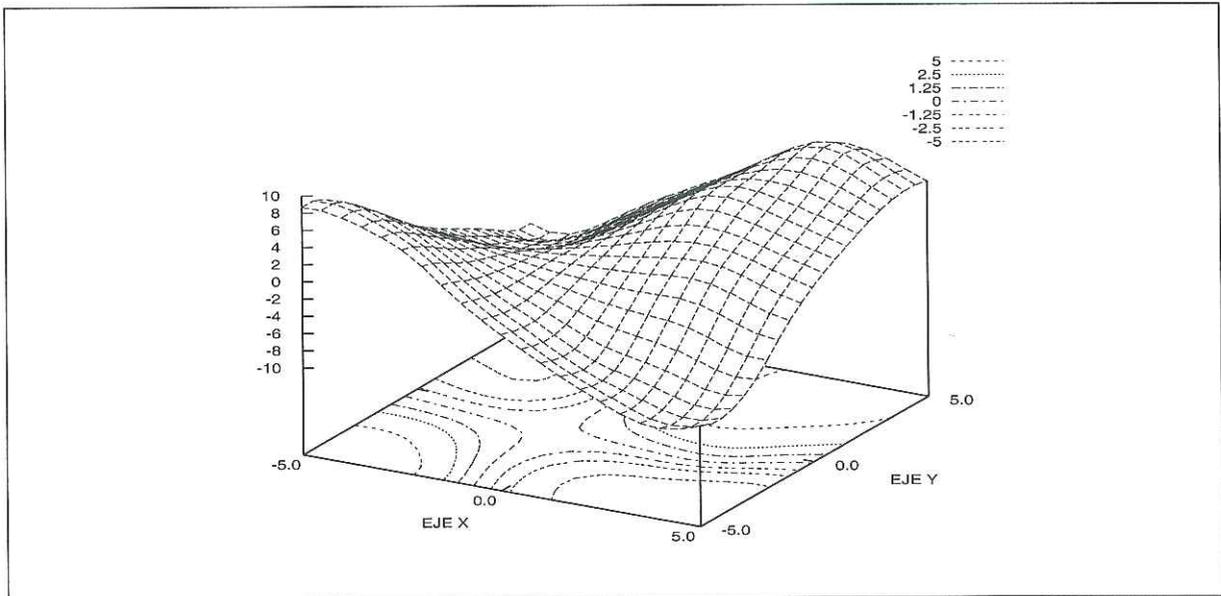


Figura 20: Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 5.0% de error

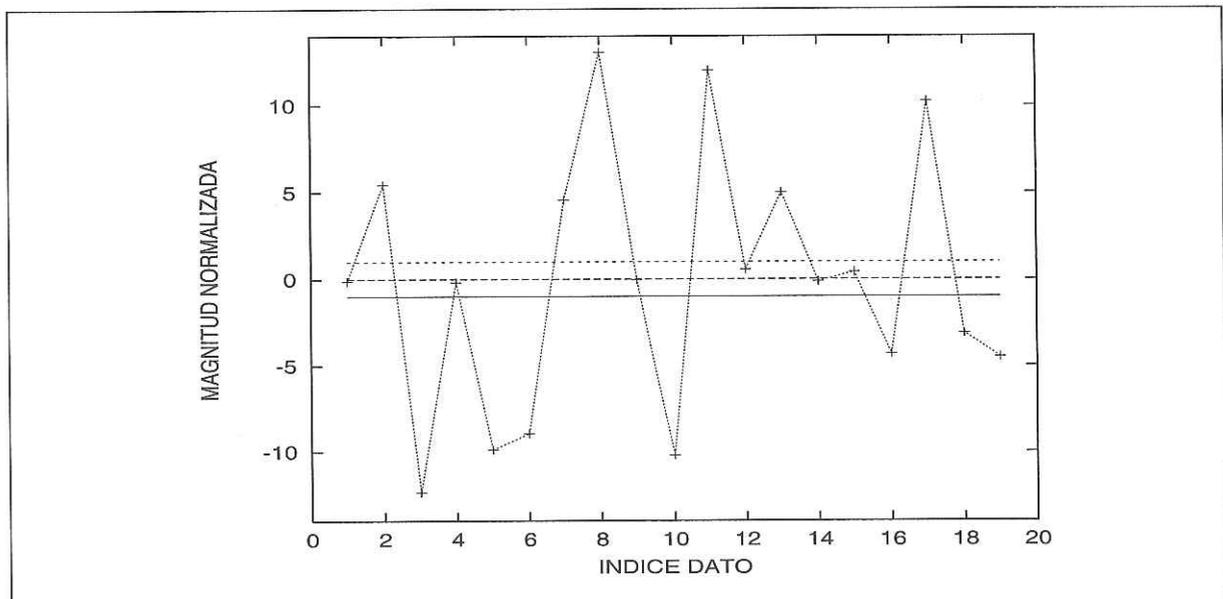


Figura 21: Gráfica de residuales para $\alpha = 1.0$; 5.0% de error

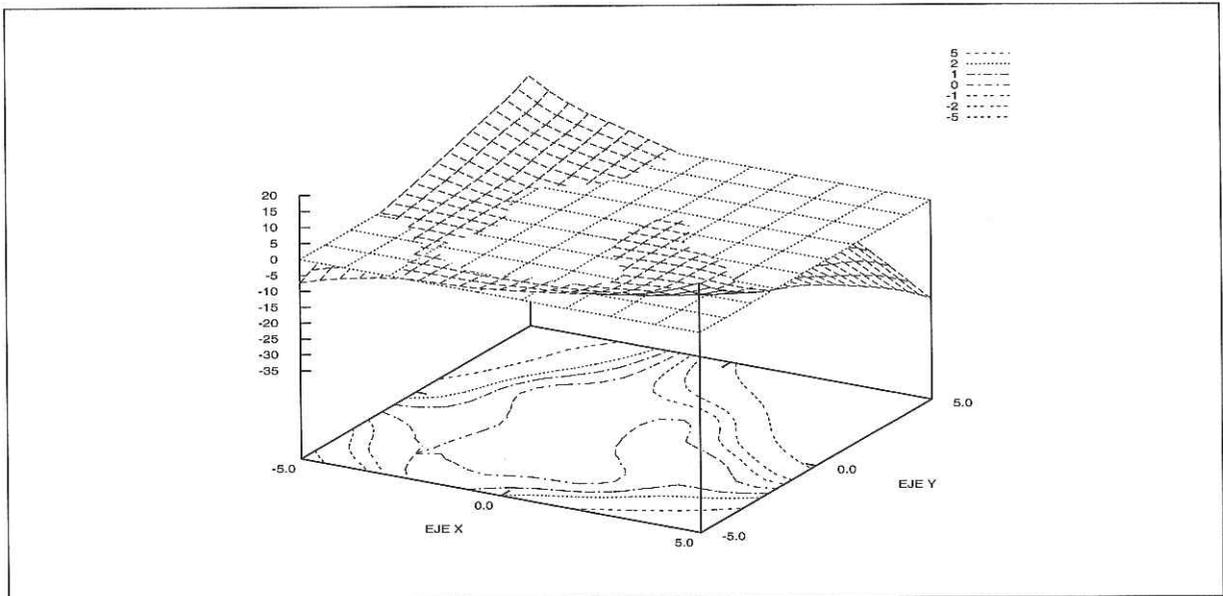


Figura 22: Gráfica del error para $\alpha = 1.0$; 5.0% de error

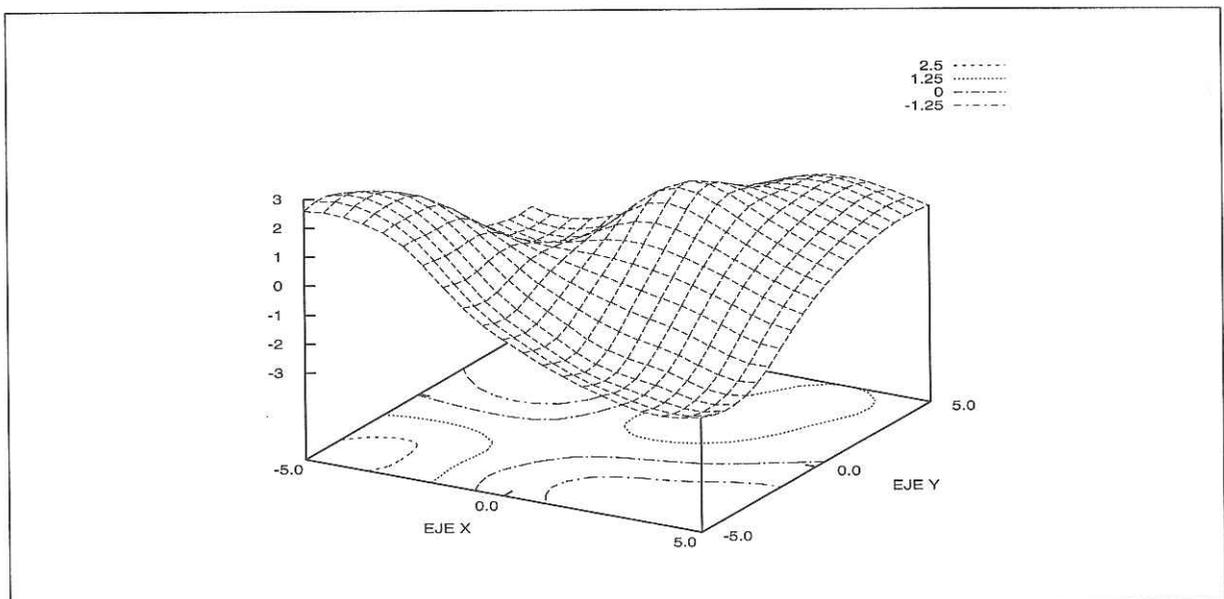


Figura 23: Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 5.0% de error

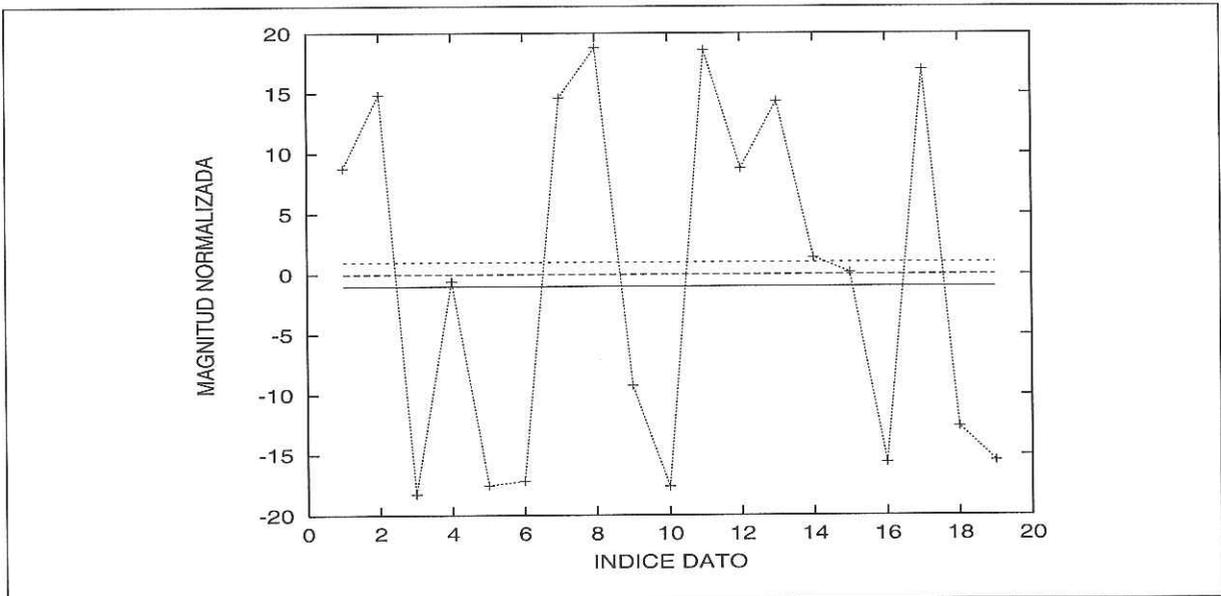


Figura 24: Gráfica de residuales para $\alpha = 10.0$; 5.0% de error

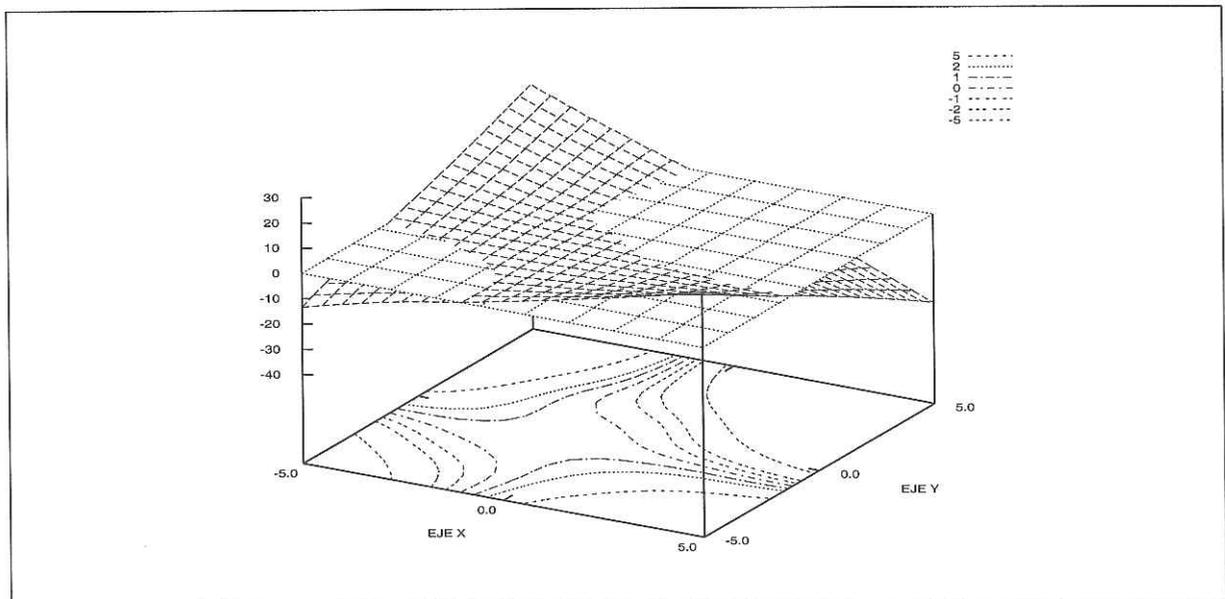


Figura 25: Gráfica del error para $\alpha = 10.0$; 5.0% de error

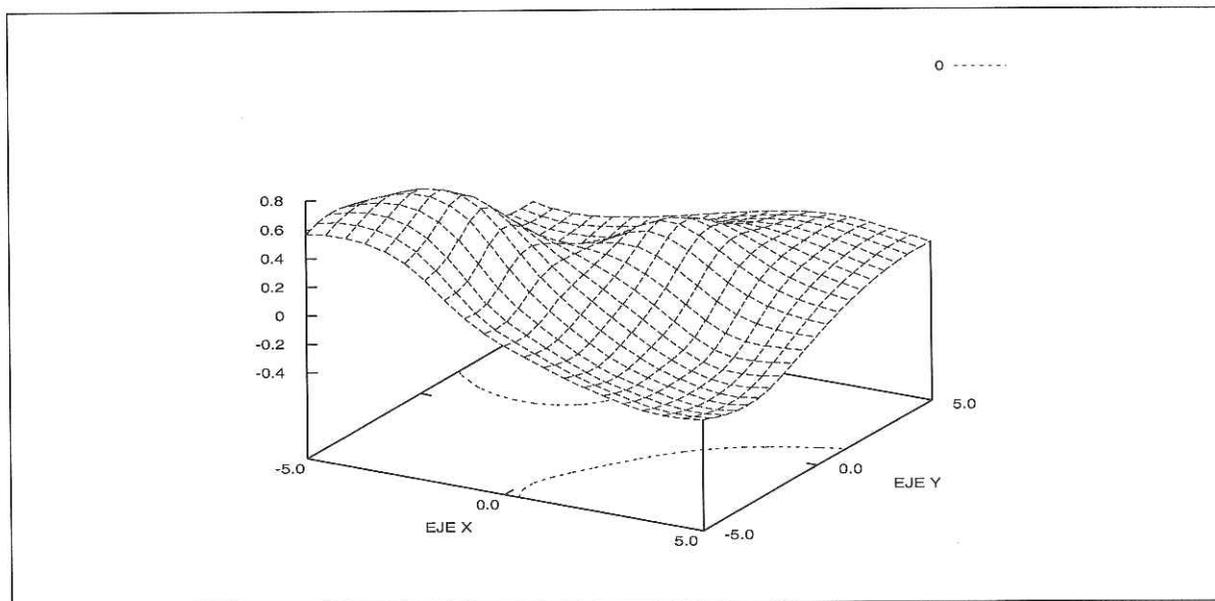


Figura 26: Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 5.0% de error

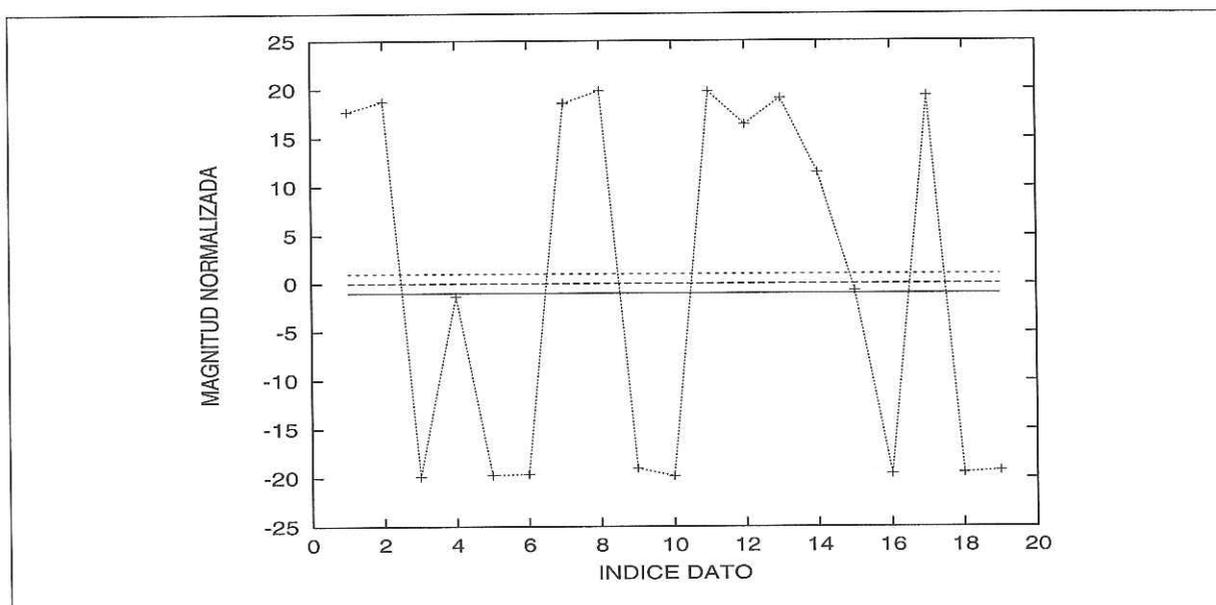


Figura 27: Gráfica de residuales para $\alpha = 100.0$; 5.0% de error

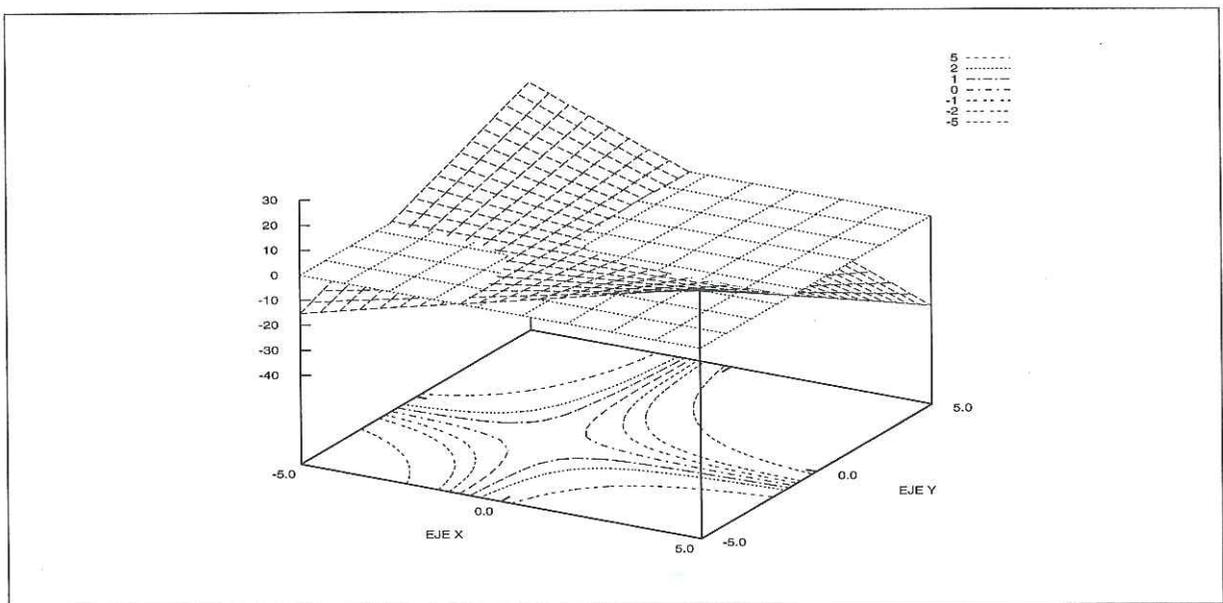


Figura 28: Gráfica del error para $\alpha = 100.0$; 5.0% de error

Los resultados para el 1.0% de error se ilustran desde la figura 29 a la 46.

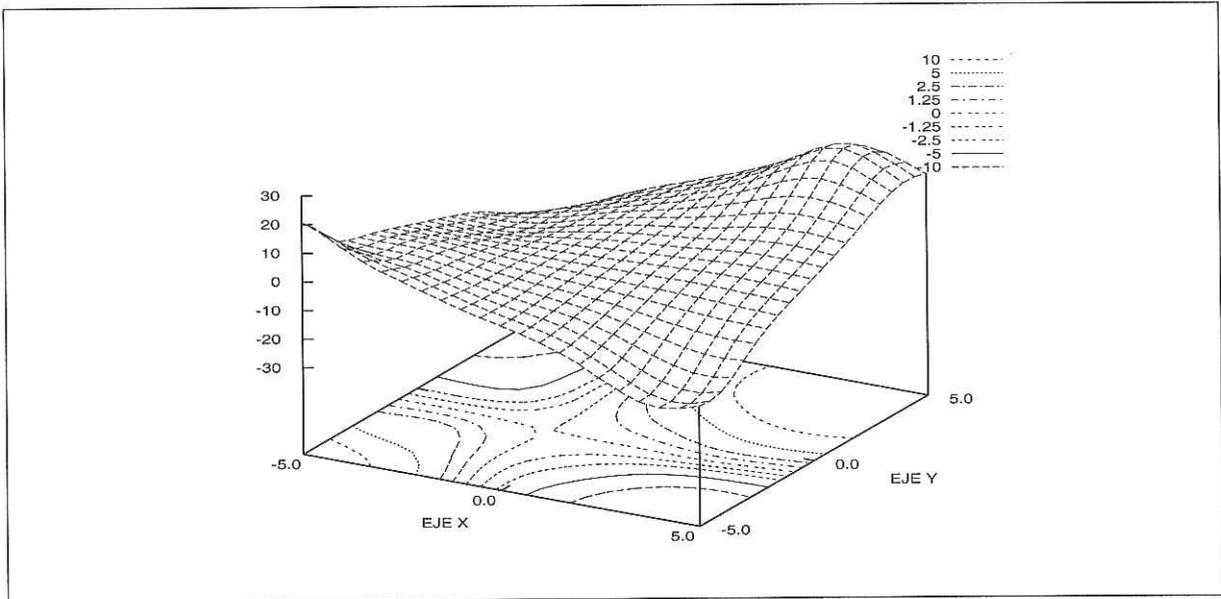


Figura 29: Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 1.0% de error

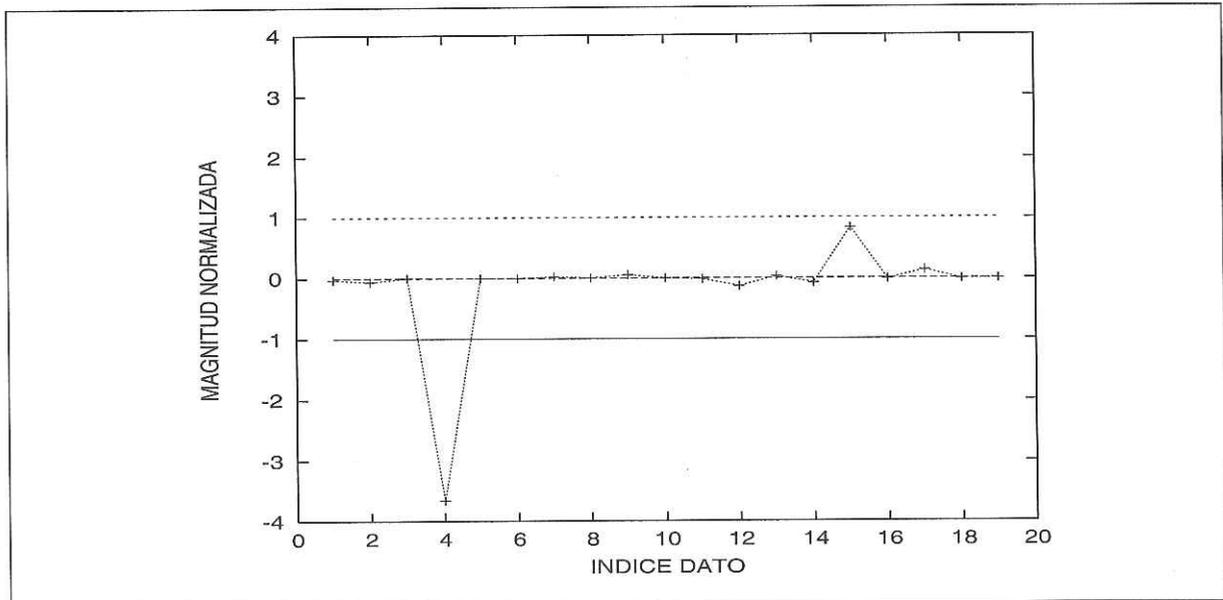


Figura 30: Gráfica de residuales para $\alpha = 0.00001$; 1.0% de error

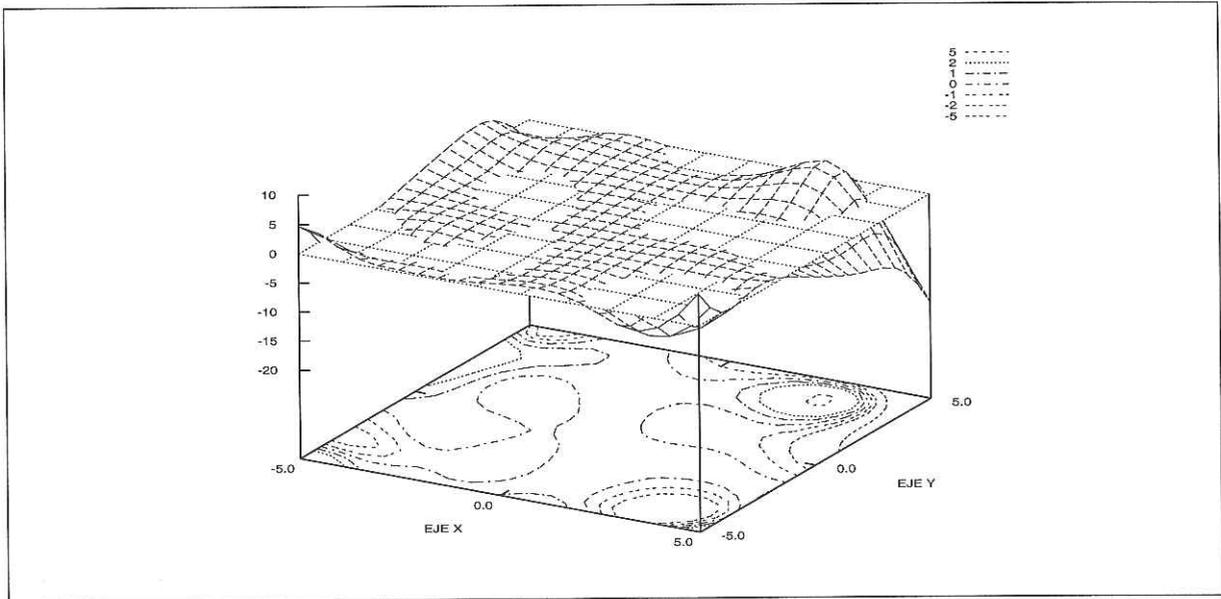


Figura 31: Gráfica del error para $\alpha = 0.00001$; 1.0% de error

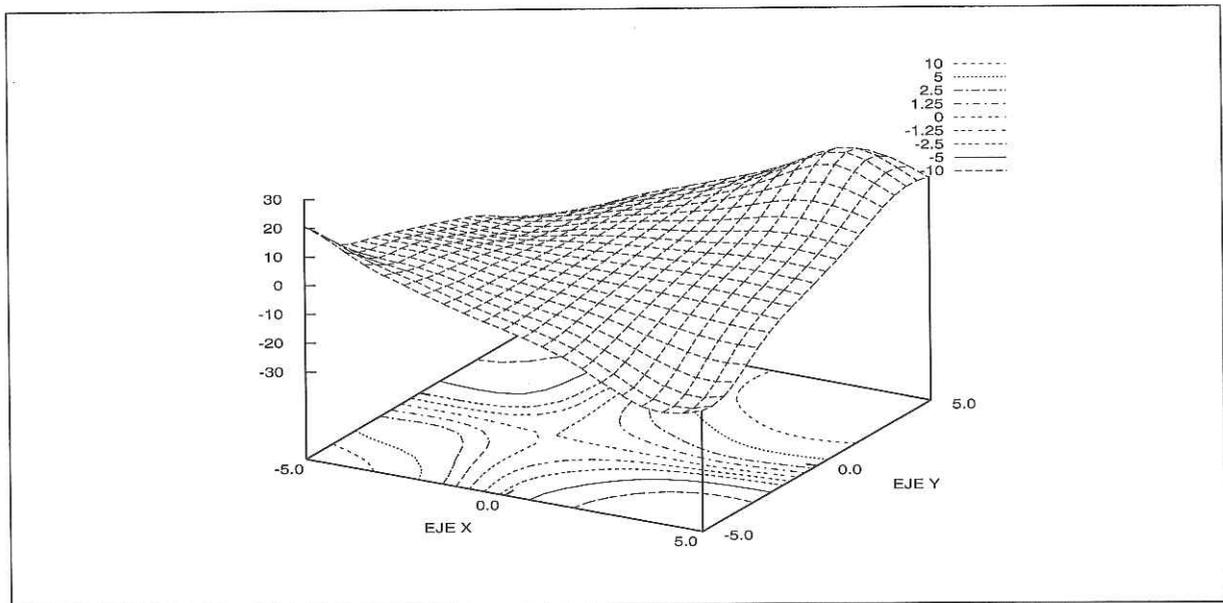


Figura 32: Gráfica de $\hat{m}(r)$ para $\alpha = 0.01$; 1.0% de error

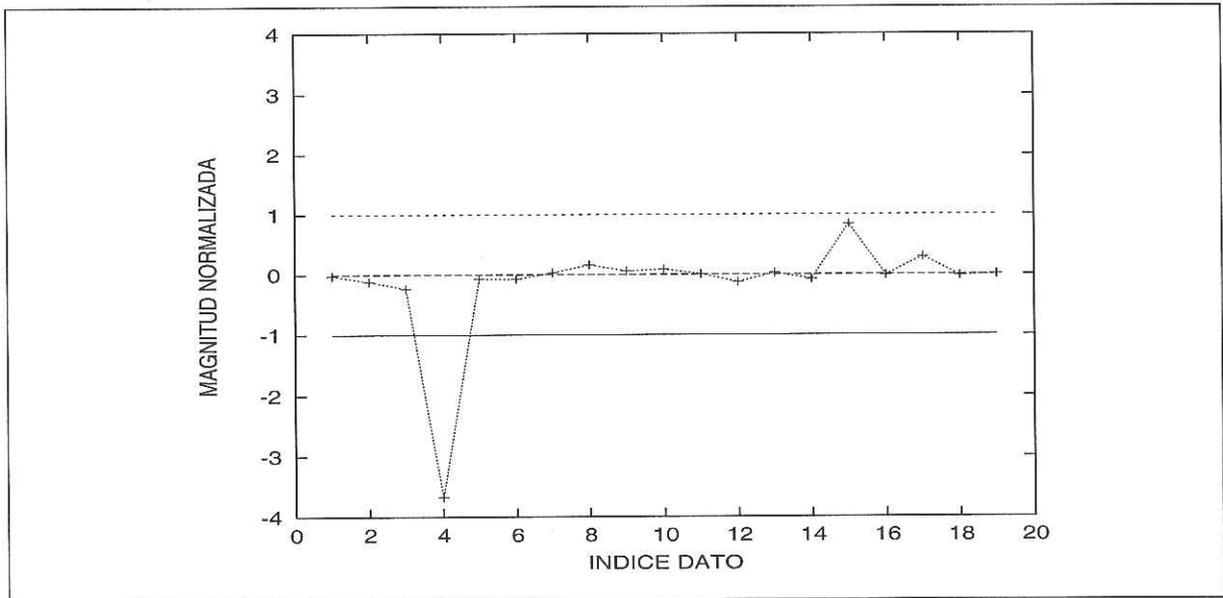


Figura 33: Gráfica de residuales para $\alpha = 0.01$; 1.0% de error

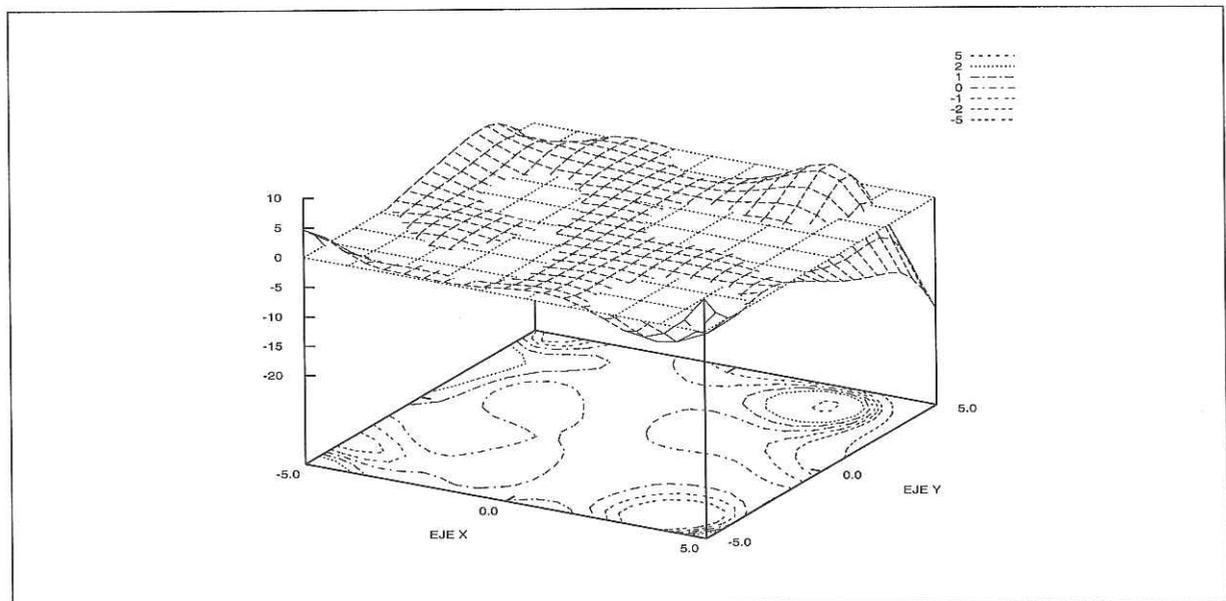


Figura 34: Gráfica del error para $\alpha = 0.01$; 1.0% de error

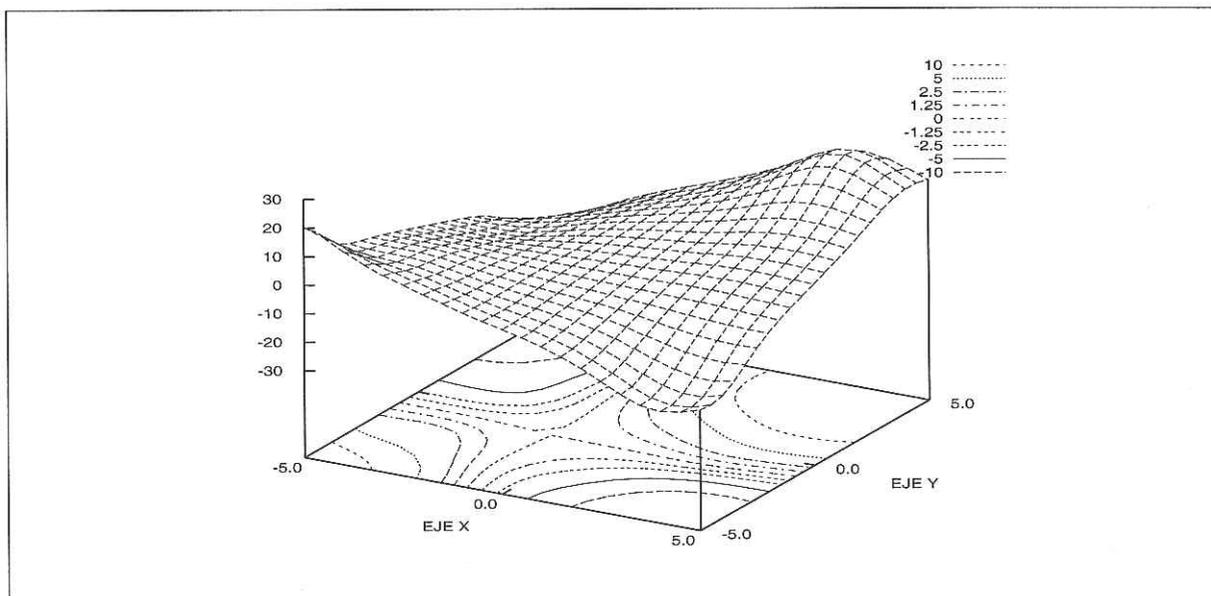


Figura 35: Gráfica de $\hat{m}(r)$ para $\alpha = 0.1$; 1.0% de error

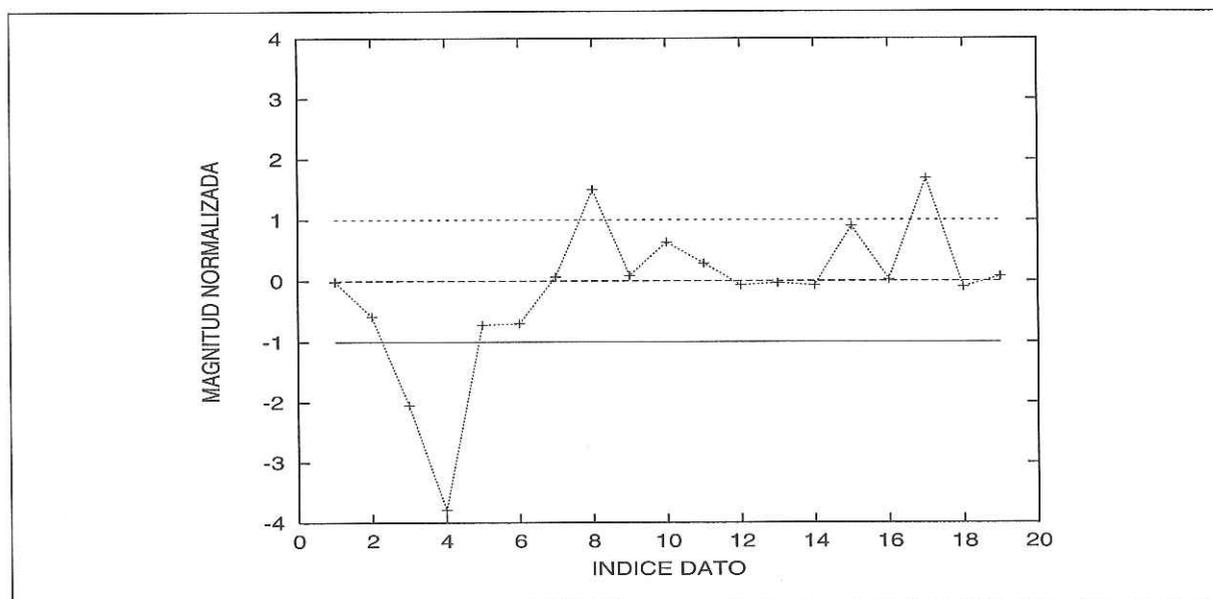


Figura 36: Gráfica de residuales para $\alpha = 0.1$; 1.0% de error

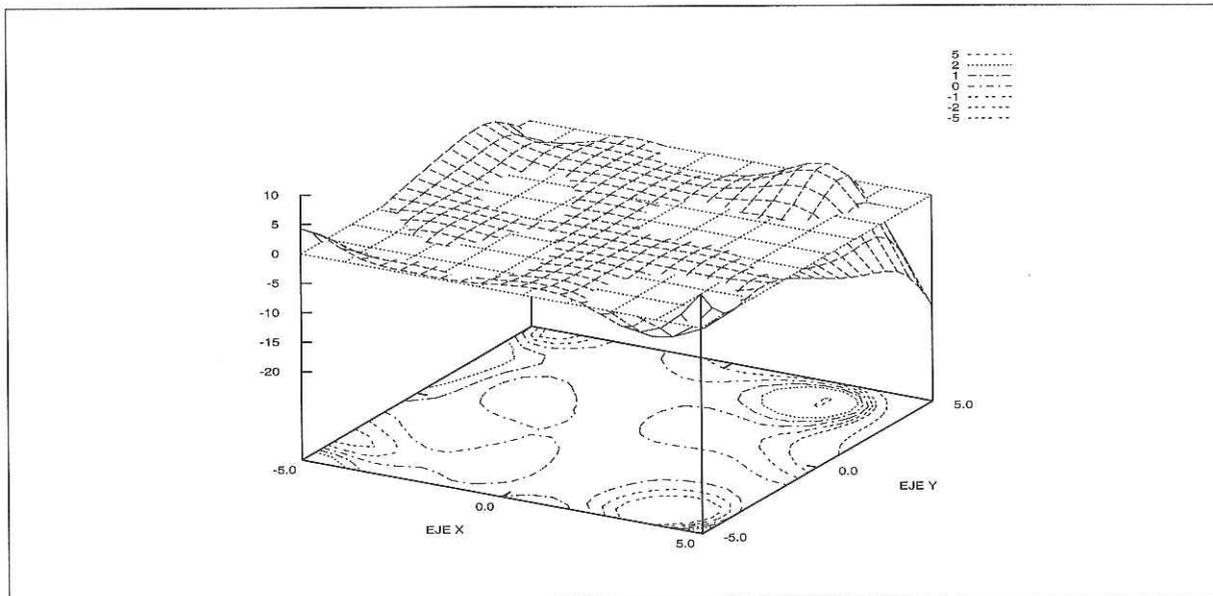


Figura 37: Gráfica del error para $\alpha = 0.1$; 1.0% de error

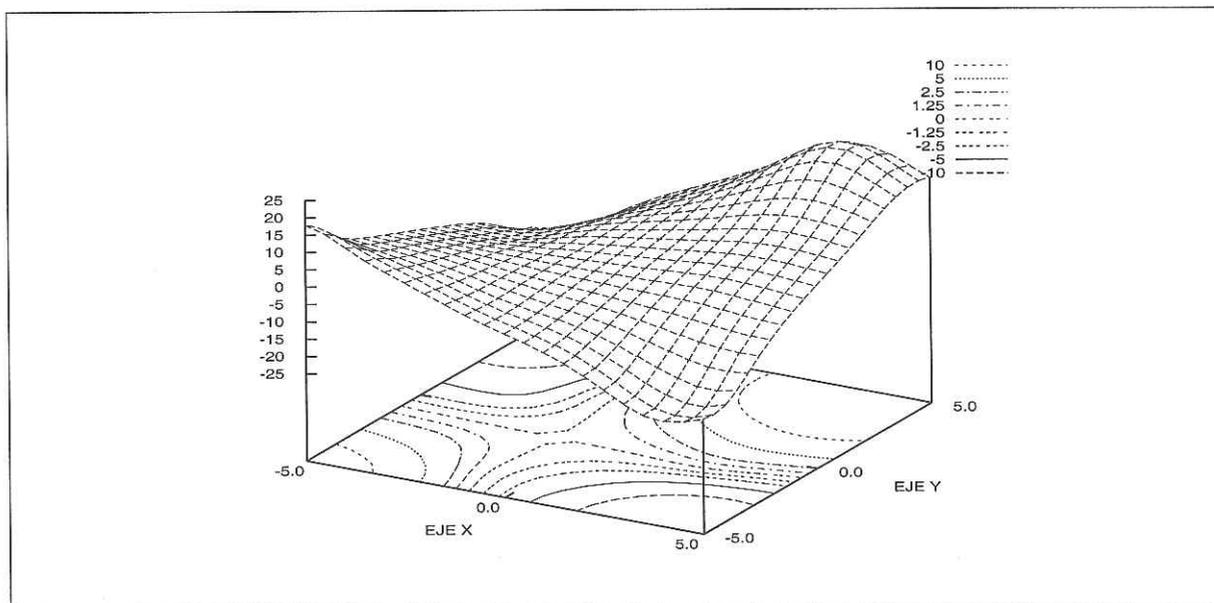


Figura 38: Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 1.0% de error

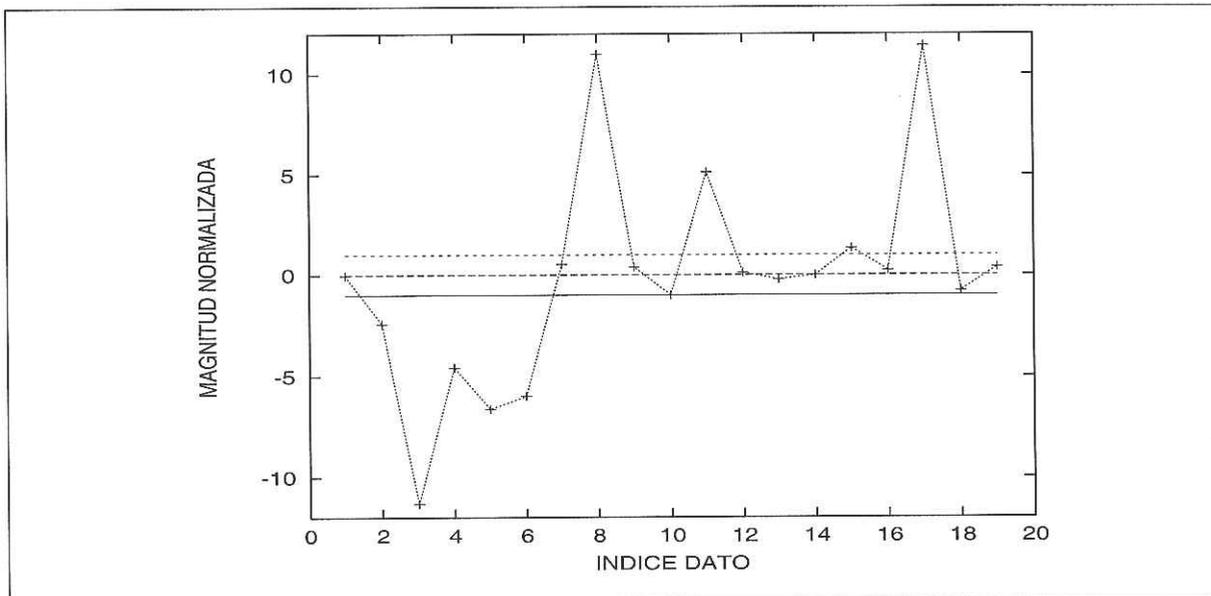


Figura 39: Gráfica de residuales para $\alpha = 1.0$; 1.0% de error

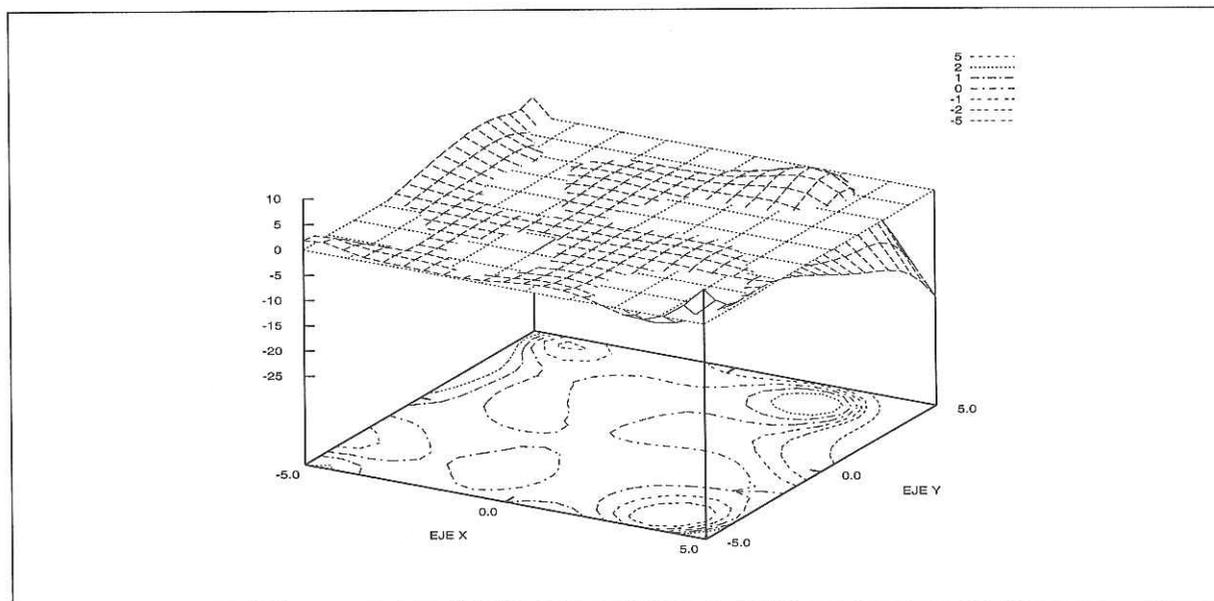


Figura 40: Gráfica del error para $\alpha = 1.0$; 1.0% de error

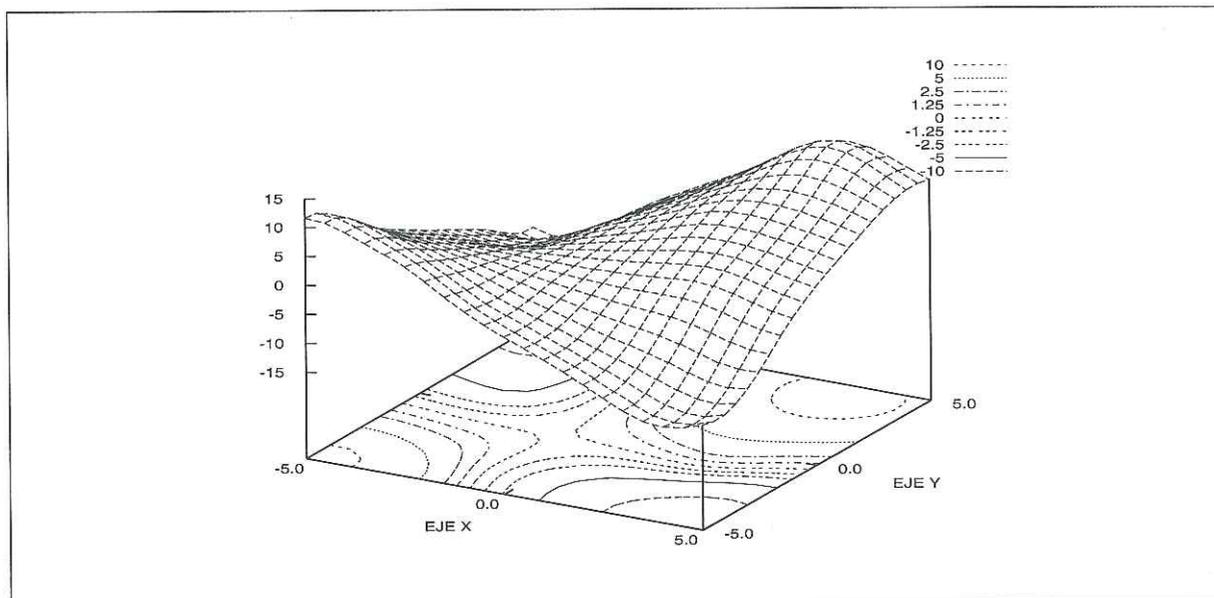


Figura 41: Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 1.0% de error

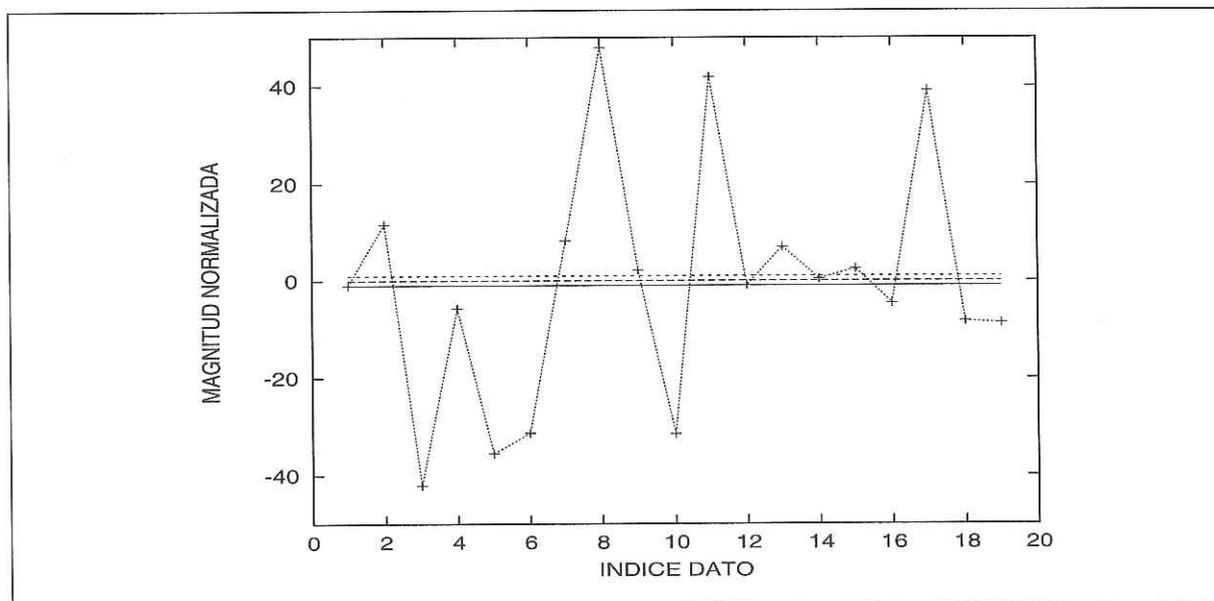


Figura 42: Gráfica de residuales para $\alpha = 10.0$; 1.0% de error

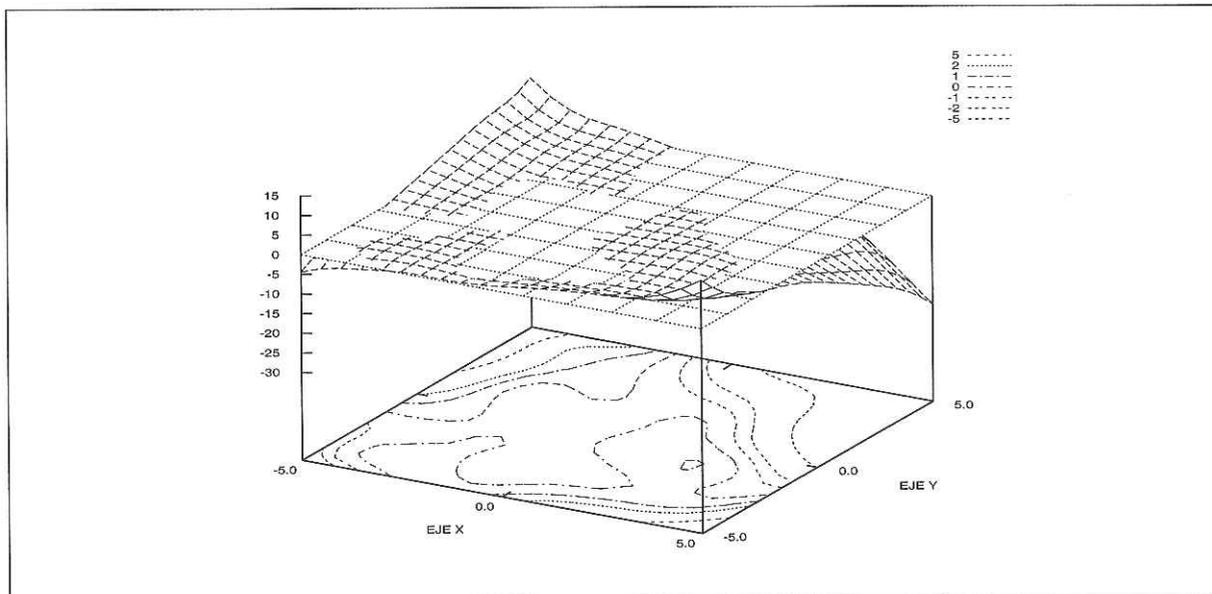


Figura 43: Gráfica del error para $\alpha = 10.0$; 1.0% de error

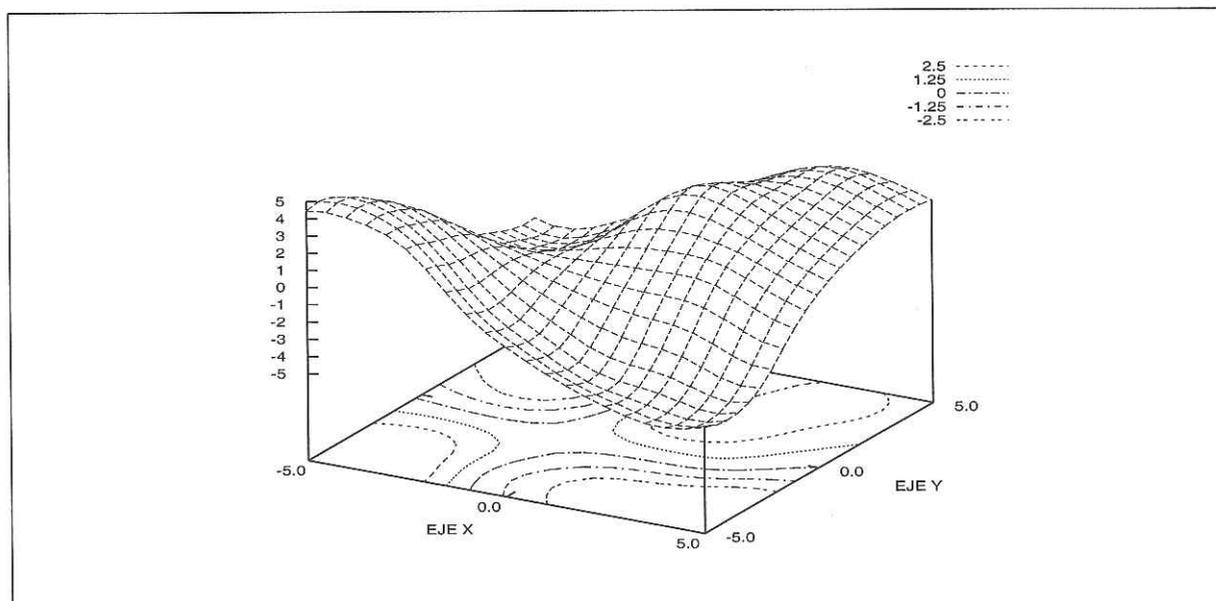


Figura 44: Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 1.0% de error

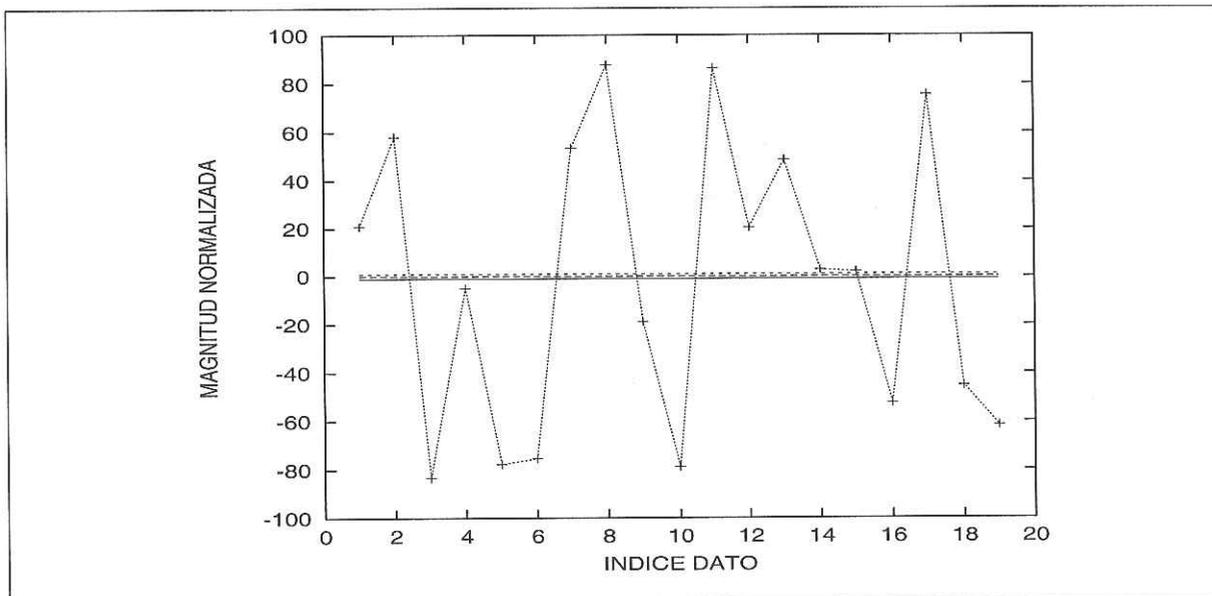


Figura 45: Gráfica de residuales para $\alpha = 100.0$; 1.0% de error

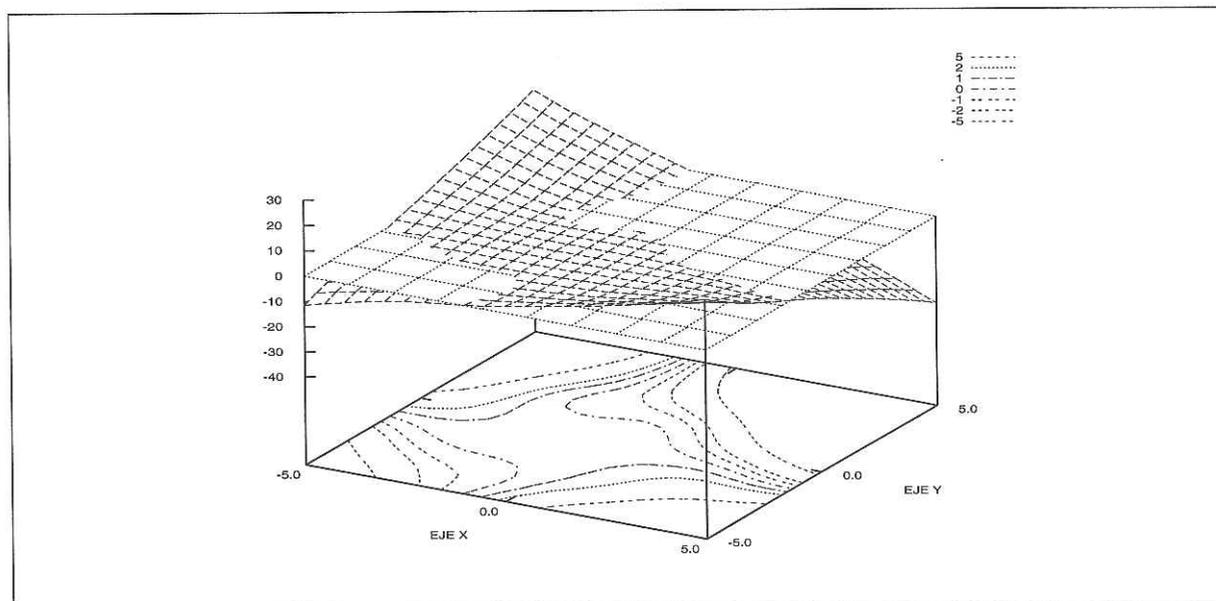


Figura 46: Gráfica del error para $\alpha = 100.0$; 1.0% de error

Los resultados para el 0.2% de error se ilustran en las figuras 47 a la 61.

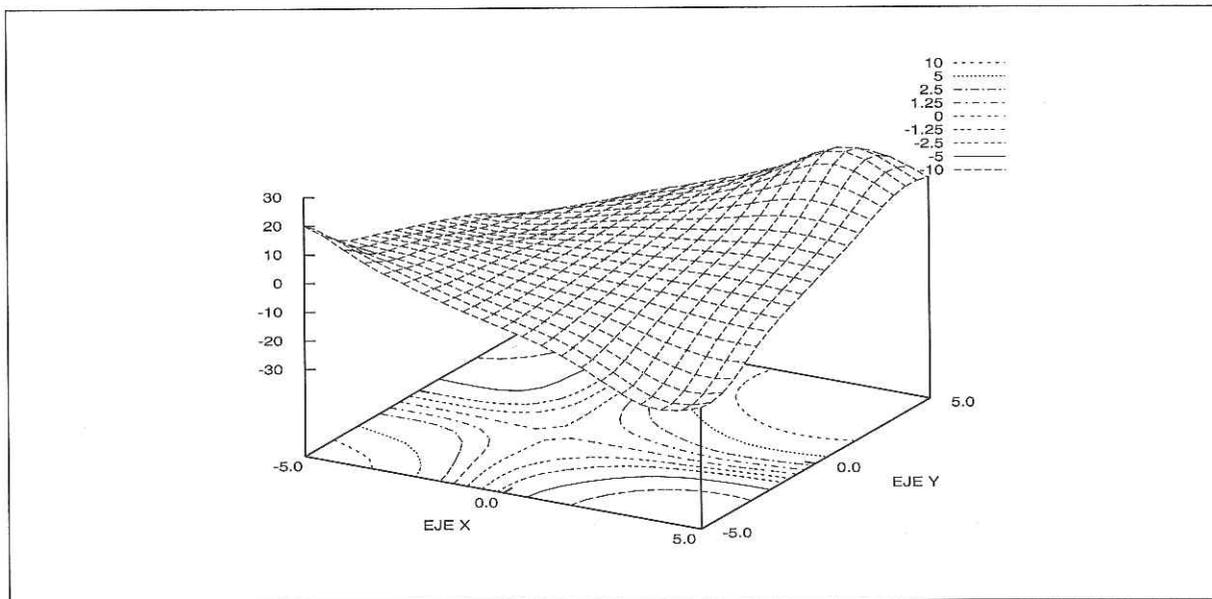


Figura 47: Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 0.2% de error

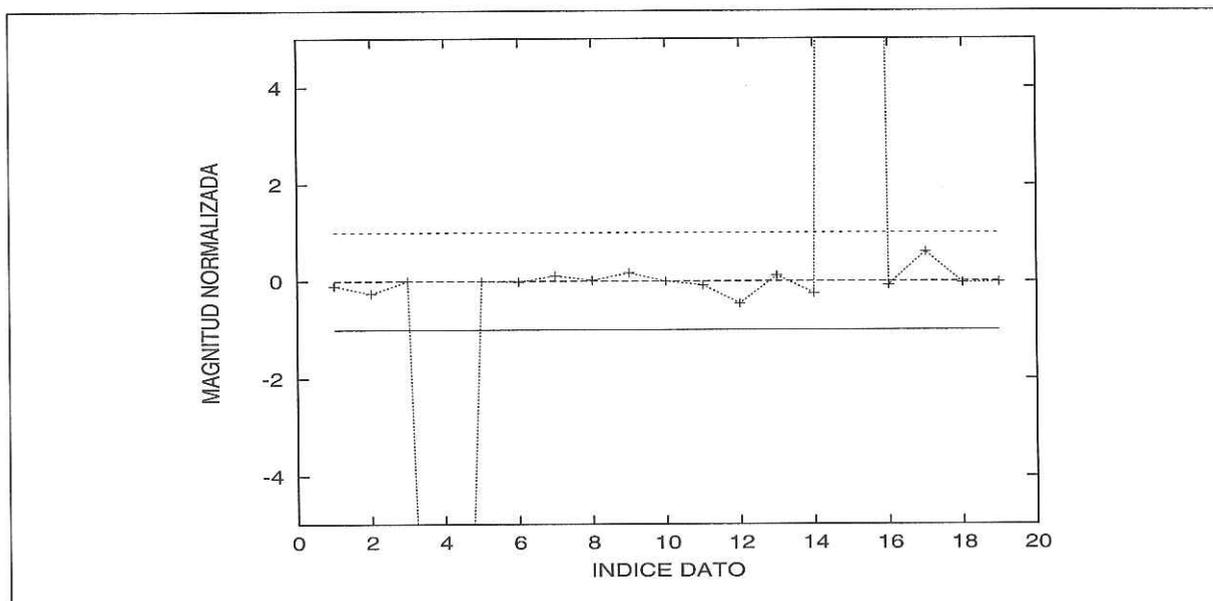


Figura 48: Gráfica de residuales para $\alpha = 0.00001$; 0.2% de error

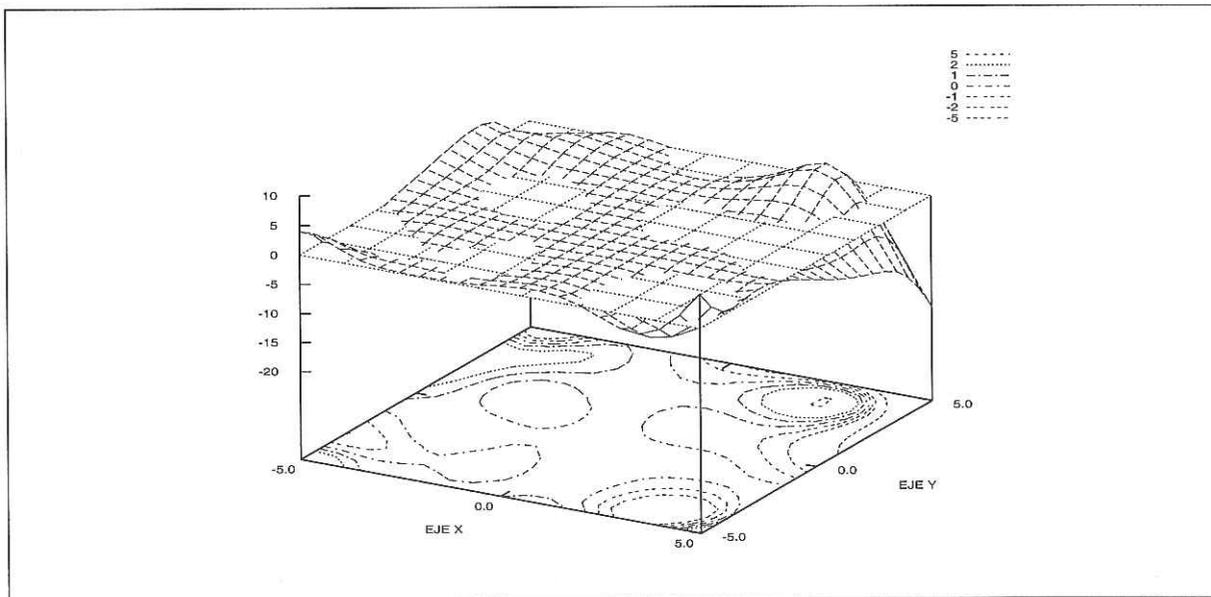


Figura 49: Gráfica del error para $\alpha = 0.00001$; 0.2% de error

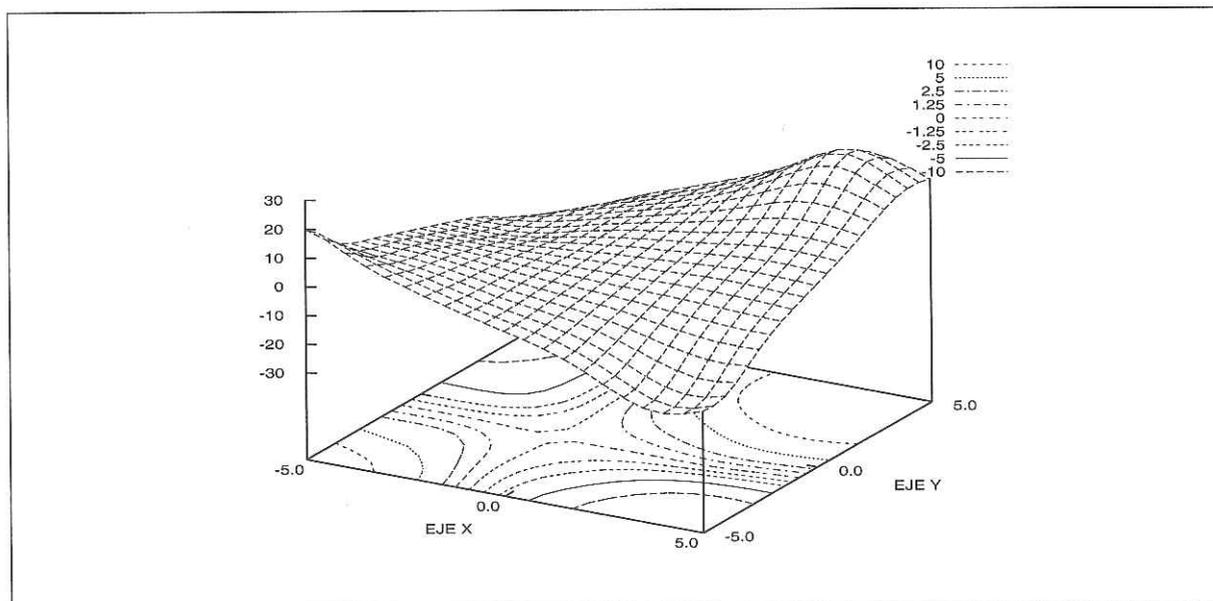


Figura 50: Gráfica de $\hat{m}(r)$ para $\alpha = 0.1$; 0.2% de error

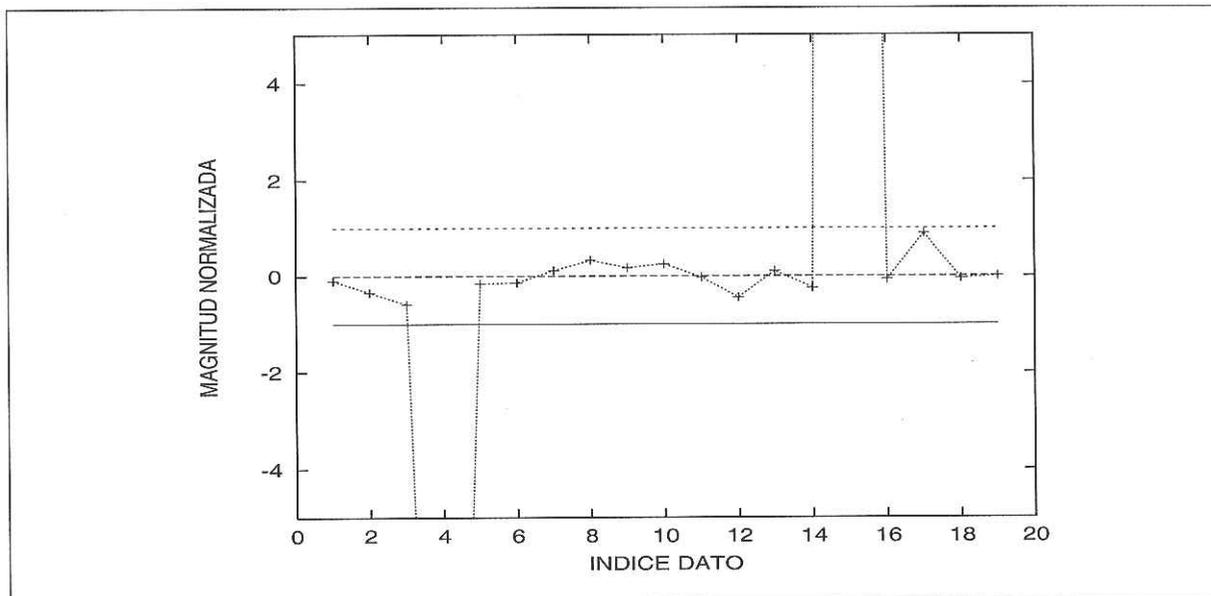


Figura 51: Gráfica de residuales para $\alpha = 0.1$; 0.2% de error

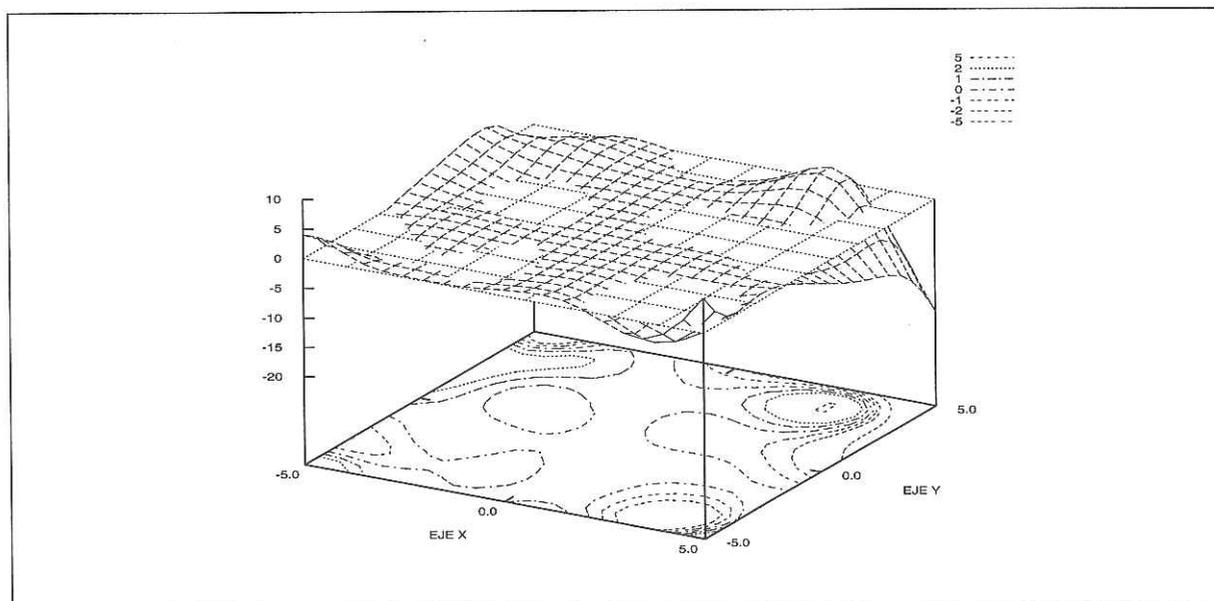


Figura 52: Gráfica del error para $\alpha = 0.1$; 0.2% de error

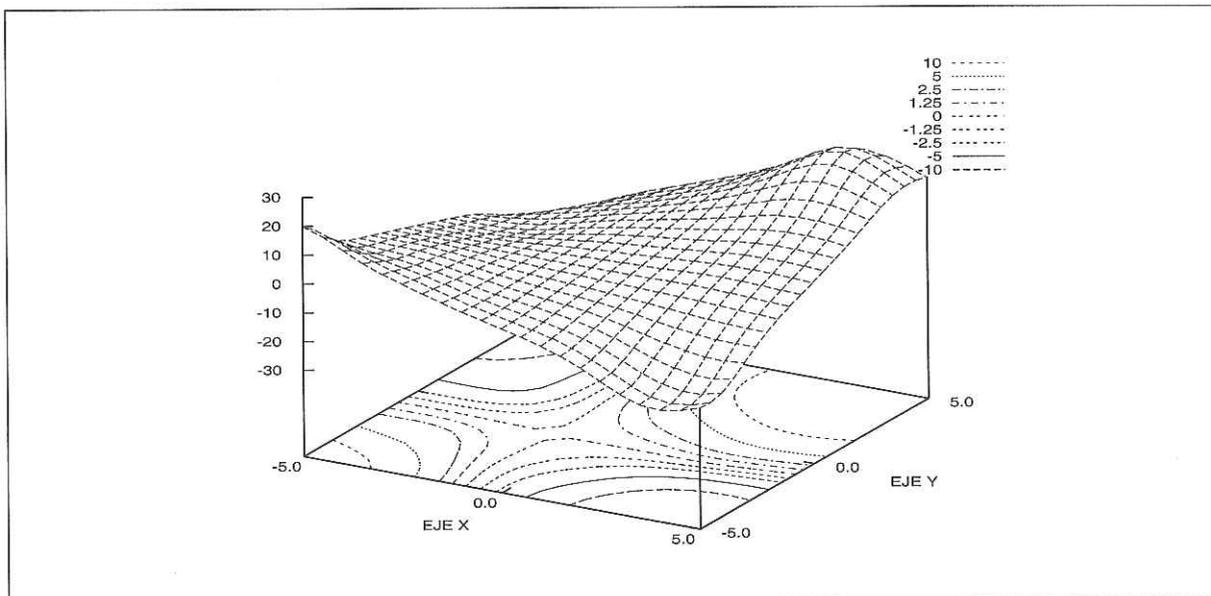


Figura 53: Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 0.2% de error

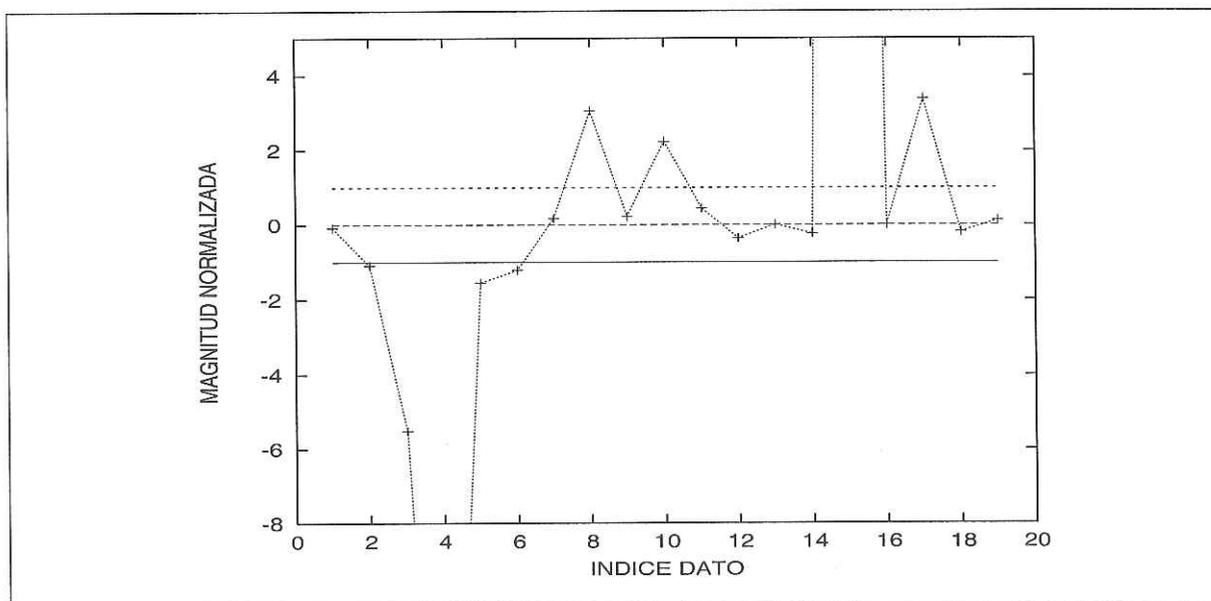


Figura 54: Gráfica de residuales para $\alpha = 1.0$; 0.2% de error

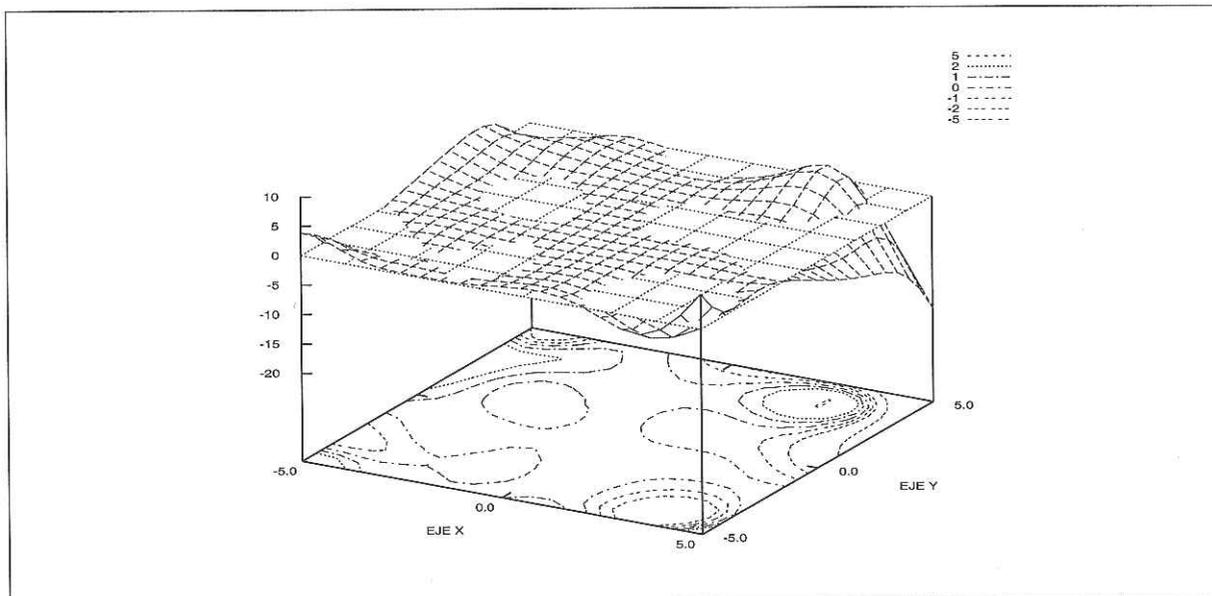


Figura 55: Gráfica del error para $\alpha = 1.0$; 0.2% de error

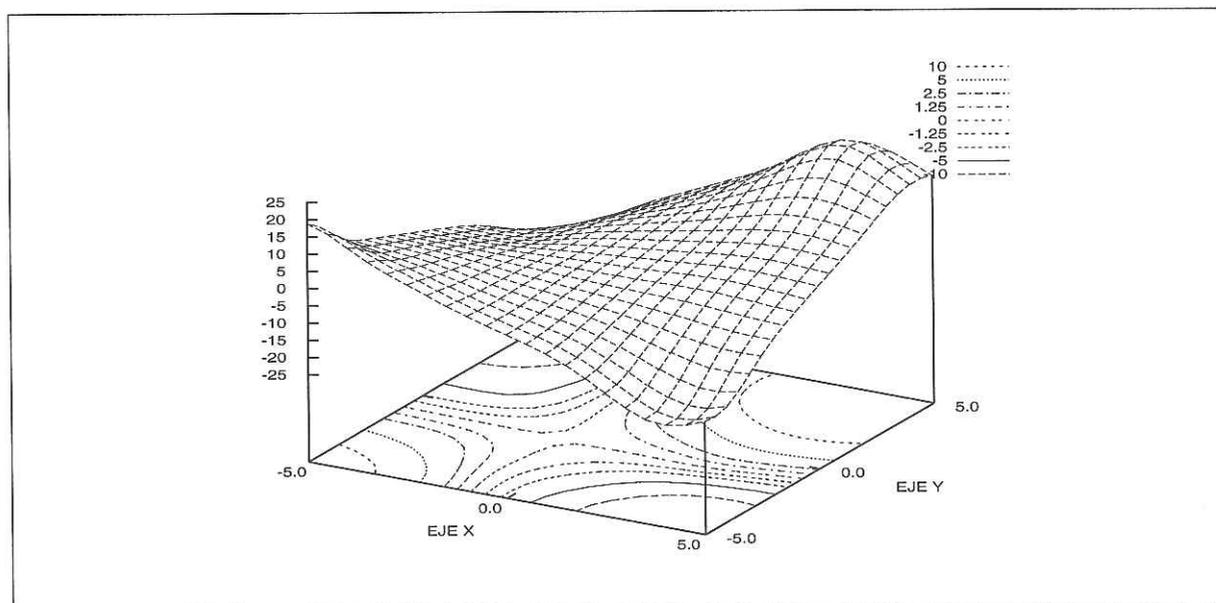


Figura 56: Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 0.2% de error

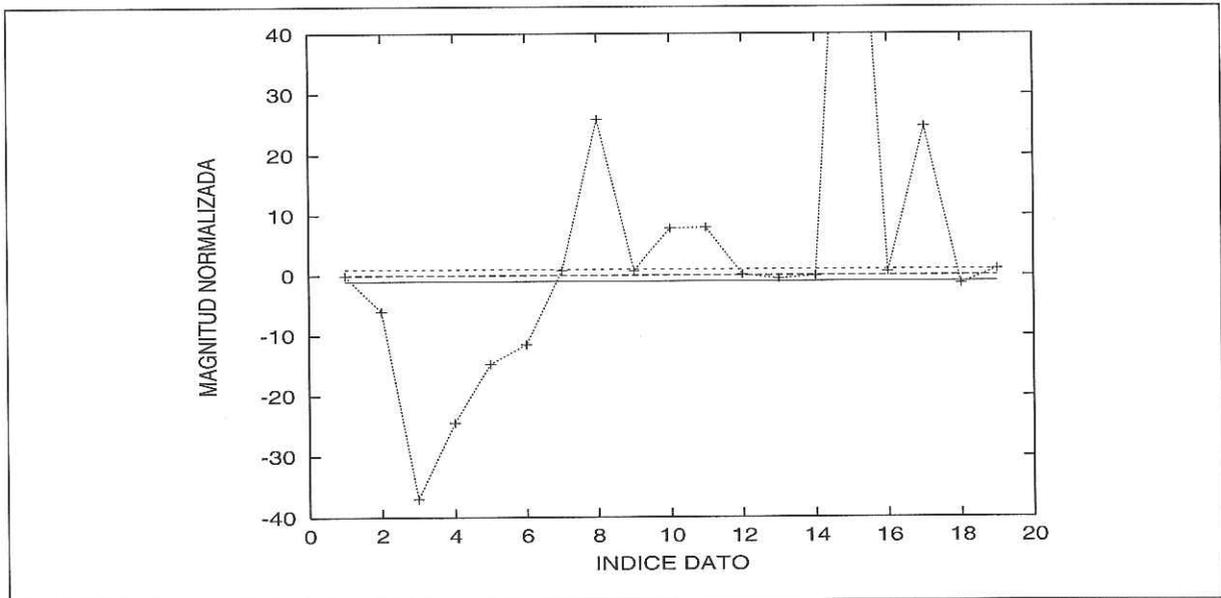


Figura 57: Gráfica de residuales para $\alpha = 10.0$; 0.2% de error

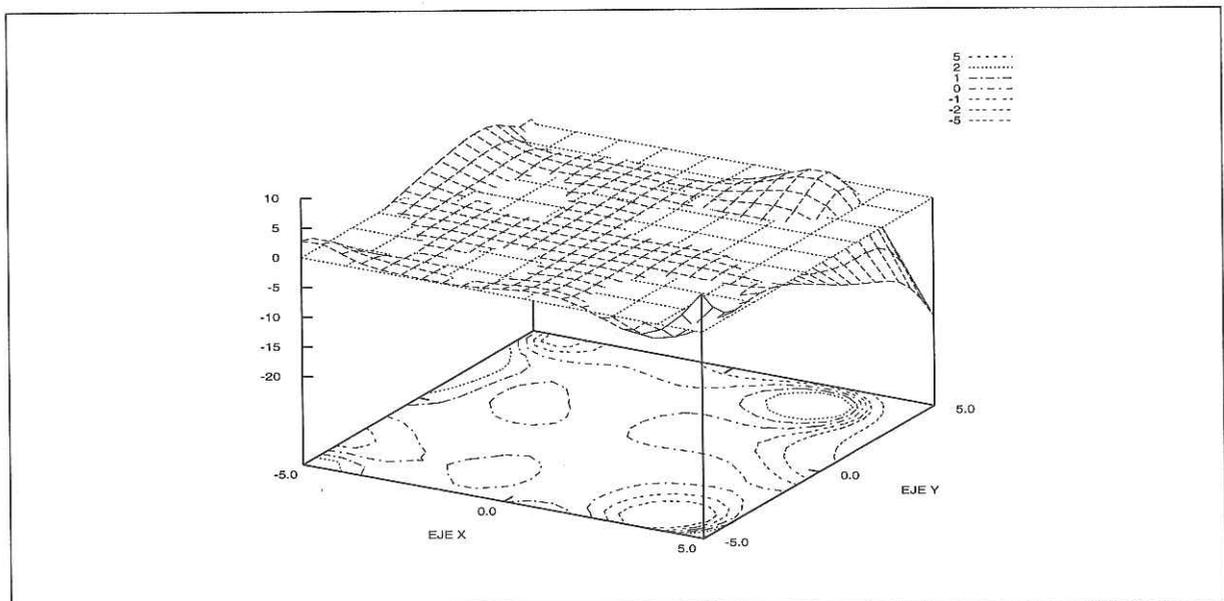


Figura 58: Gráfica del error para $\alpha = 10.0$; 0.2% de error

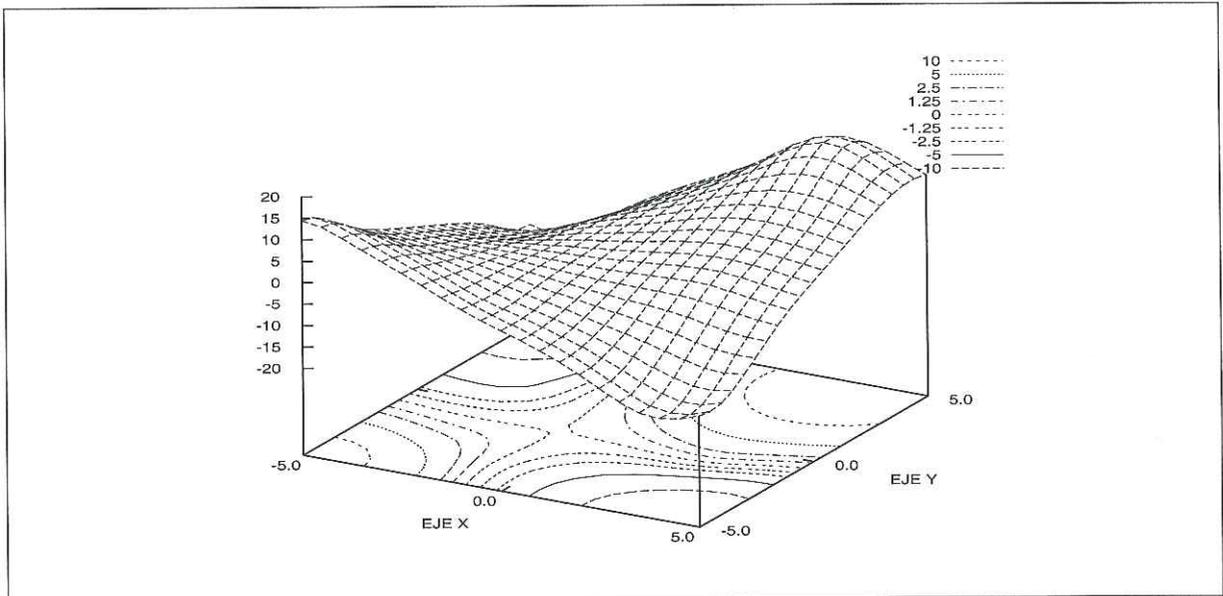


Figura 59: Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 0.2% de error

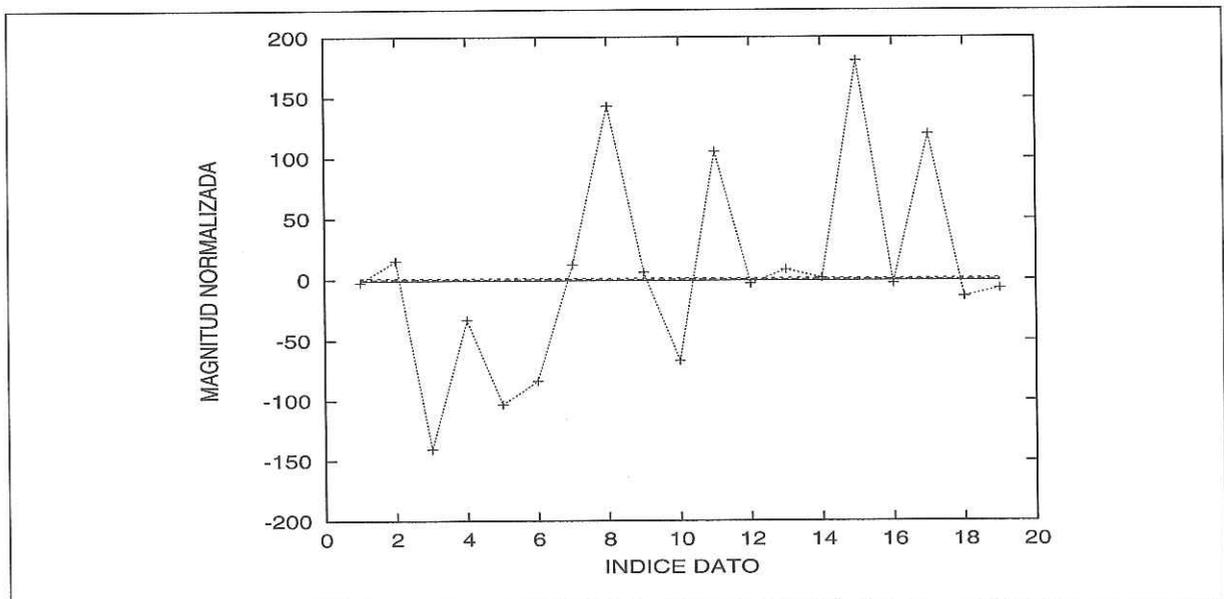


Figura 60: Gráfica de residuales para $\alpha = 100.0$; 0.2% de error

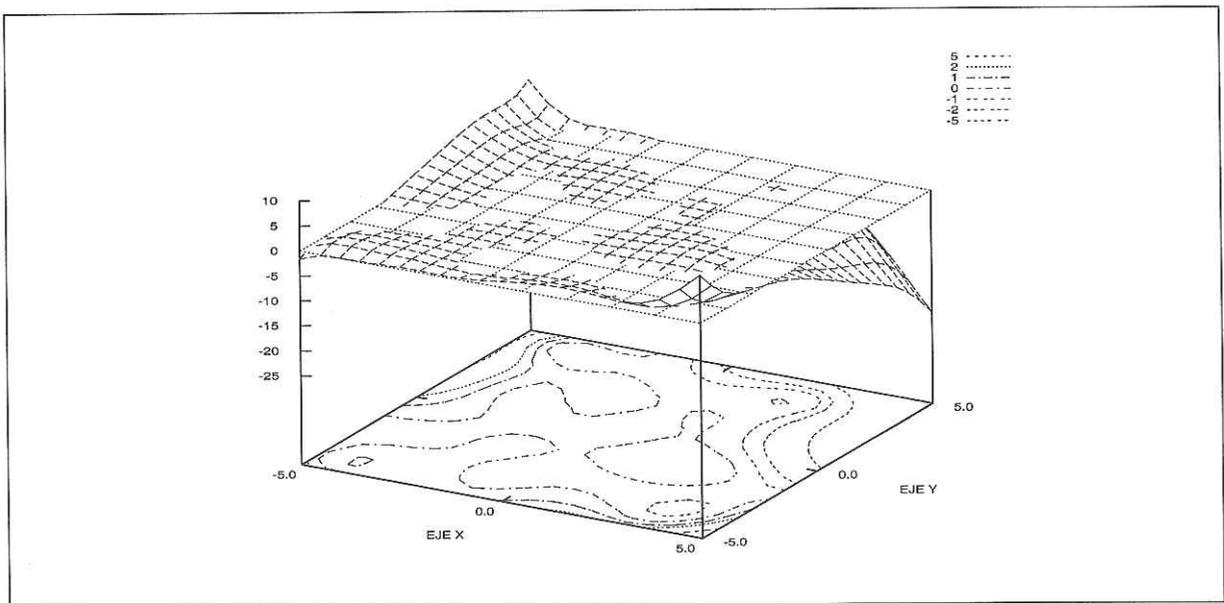


Figura 61: Gráfica del error para $\alpha = 100.0$; 0.2% de error

II.8 Análisis de Resultados. Conclusiones

Los resultados obtenidos para los distintos niveles de ruido en los datos muestran el siguiente comportamiento.

Para valores pequeños del parámetro α ($\approx 10^{-4}$) se observa:

- La estimación de la solución muestra oscilaciones grandes.
- Los errores son casi del mismo orden que el valor de la solución verdadera.
- Los residuales son muy pequeños y despreciables.

Para valores grandes del parámetro α ($\approx 10^2$) se observa:

- La estimación tiende al plano $z = 0$, con algunas oscilaciones.
- Los errores, por lo tanto, son casi del 100%, es decir, corresponde a la solución verdadera con signo contrario debido a la definición del error (solución estimada - solución verdadera).
- Los componentes del vector de residuales son muy grandes y muestran conductas sistemáticas, ya que no sólo transmiten el error observacional sino aseguran error por sobreestabilización.

Analizando residuales, error de la estimación y estabilidad de la estimación, las estimaciones óptimas que se eligen, para cada nivel de ruido, son las siguientes:

a) Para datos con el 0.2% de ruido la solución óptima estimada corresponde a

$$\alpha = 1.0$$

b) Para datos con el 1.0% de ruido la solución óptima estimada corresponde a

$$\alpha = 0.1$$

c) Para datos con el 5.0% de ruido la solución óptima estimada corresponde a

$$\alpha = 0.1$$

En todos los casos, la solución estimada contiene oscilaciones que no pertenecen a la solución verdadera; al mismo tiempo, se encuentran residuales altos. Estas diferencias parecen deberse al número pequeño de puntos de observación y al hecho de que el área de interés no es cubierta uniformemente por las observaciones, mostrando huecos en los cuales no se hacen observaciones. Aún así, el procedimiento numérico computacional recupera adecuadamente la función incógnita, lo que nos lleva a concluir que el procedimiento no contiene errores y puede pasar a la etapa siguiente en su aplicación.

Las tendencias observadas en las soluciones estimadas, para valores pequeños y grandes de α están de acuerdo con lo previsto por la expresión analítica del estimador, es decir, valores pequeños de α nos llevan a una solución mínimo cuadrática inestable mientras que el incremento de α hace tender la solución estimada a la solución nula, con predominio de la estabilidad.

Capítulo III

Aplicación del Procedimiento de Inversión

En este capítulo se resuelve un problema que representa mayor dificultad en su solución que el problema analizado y resuelto en el capítulo anterior.

III.1 Definición del Problema

El kernel utilizado para este problema es el mismo que se utilizó en el problema de validación, una función gaussiana bivalente

$$K_i(x, y) = e^{-(x-x_i)^2} e^{-(y-y_i)^2} \quad (44)$$

donde las desviaciones estándar implícitas son

$$\sigma_x = \frac{1.0}{\sqrt{2.0}} \quad \sigma_y = \frac{1.0}{\sqrt{2.0}}$$

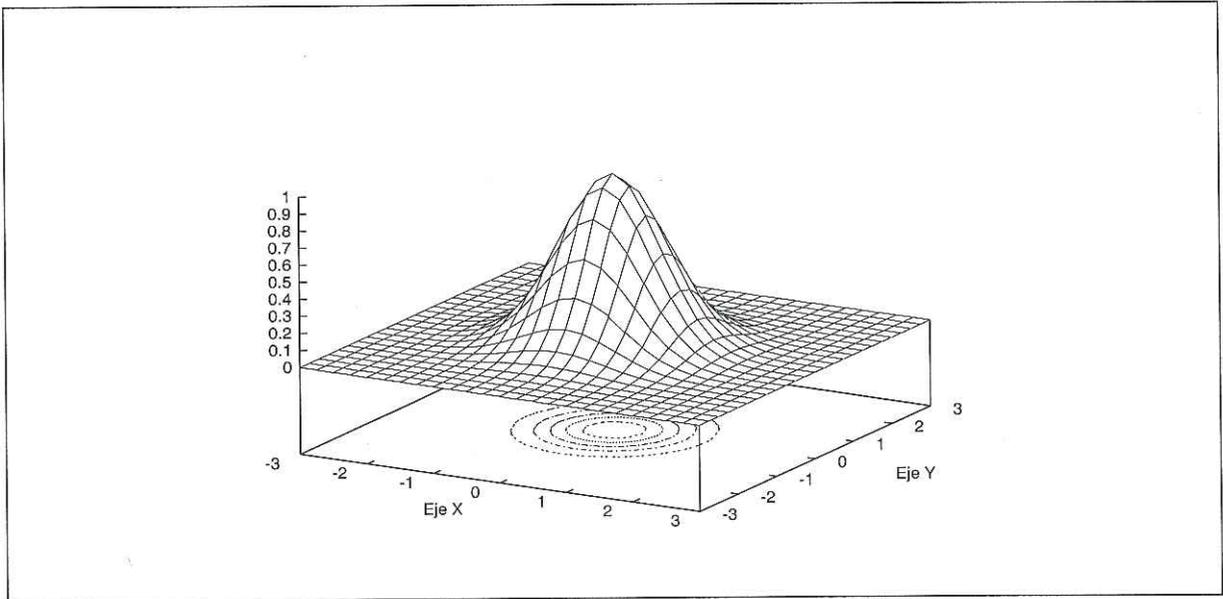


Figura 62: Gráfica de la función gaussiana (44): $x_i = 0$, $y_i = 0$

Como función incógnita $m(x, y)$ se elige una función compuesta por la suma de dos funciones gaussianas bivariantes, cada una con promedio y desviaciones (σ_x y σ_y) diferentes.

$$m(x, y) = e^{-k_1(x-x_1)^2} e^{-n_1(y-y_1)^2} + e^{-k_2(x-x_2)^2} e^{-n_2(y-y_2)^2} \quad (57)$$

donde

$$k_1 = 0.2 \quad k_2 = 0.1$$

$$n_1 = 0.4 \quad n_2 = 0.075$$

$$x_1 = -2.0 \quad x_2 = 0.5$$

$$y_1 = -2.0 \quad y_2 = 1.0$$

Los puntos (x_1, y_1) y (x_2, y_2) son los puntos donde tienen su máximo las funciones gaussianas.

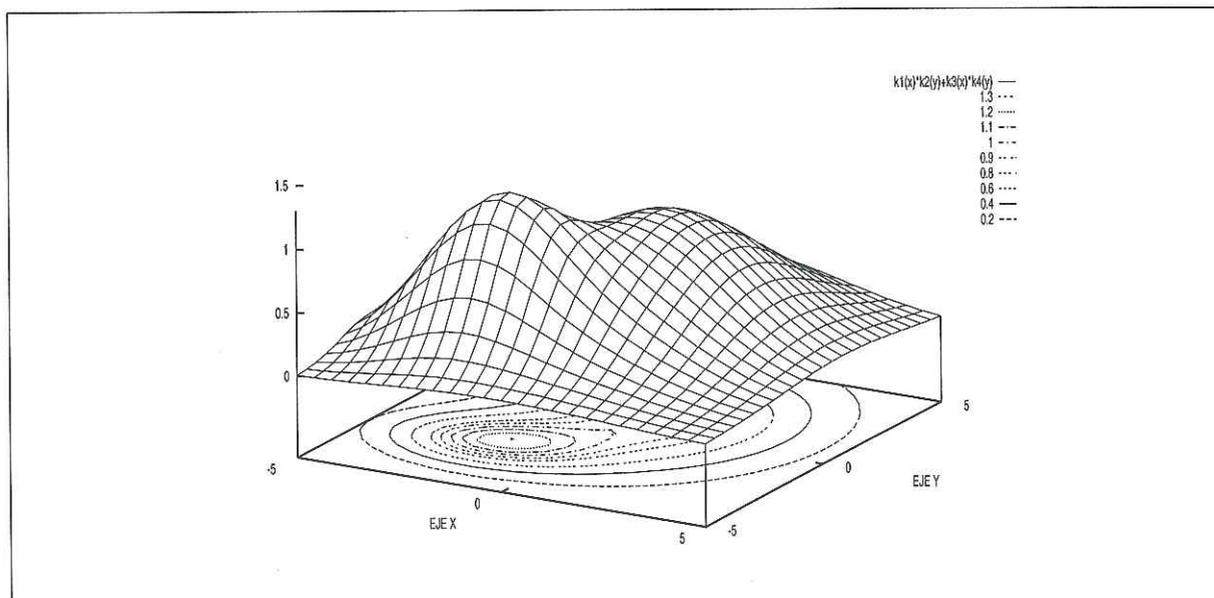


Figura 63: Gráfica de la función $m(x, y)$ utilizada en el segundo problema

Las razones para seleccionar de nuevo el kernel (44) del problema de validación son dos:

- a) No se pierde generalidad en la aplicación del procedimiento numérico computacional.
- b) El proceso de inversión computacional se simplifica, ya que se utilizan de nuevo algunos procedimientos computacionales que se utilizaron en la validación.

Las dificultades que presenta este problema, dados el kernel (44) y la función (57), son los siguientes.

- a) La integración sobre todo el plano del producto $K_i(x, y) m(x, y)$ no tiene solución analítica, por lo que el vector de datos se genera numéricamente.
- b) La forma geométrica de $m(x, y)$ es en general más complicada que la función propuesta en el capítulo anterior; la función presenta dos crestas, las cuales permiten

probar la capacidad del procedimiento de inversión para resolver ambos máximos.

III.2 Planteamiento del Problema Directo

El problema directo consiste en resolver la integral doble

$$g_i = \int K_i(r)m(r)dr \quad (38)$$

Sustituyendo las expresiones (44) y (57) tenemos

$$g_i = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x-x_i)^2} e^{-(y-y_i)^2} \left[e^{-k_1(x-x_1)^2} e^{-n_1(y-y_1)^2} + e^{-k_2(x-x_2)^2} e^{-n_2(y-y_2)^2} \right] dx dy \quad (58)$$

Esta integral no tiene solución analítica conocida por lo que se resuelve numéricamente.

Los detalles de la integración numérica se darán más adelante.

III.3 Planteamiento del Problema Inverso

Para la solución del problema inverso, dados el kernel (44) y el vector de observaciones g , cuyos elementos se calculan con la expresión (58), se estima la función $m(x, y)$ mediante la expresión (37) del capítulo I.

$$m(r) = e^T(r)(S + \alpha C_{yy})^{-1}g + \frac{1 - e^T(r)(S + \alpha C_{yy})^{-1}d}{d^T(S + \alpha C_{yy})^{-1}d} d^T(S + \alpha C_{yy})^{-1}g \quad (37)$$

Los elementos de los vectores d y $e_i(r)$, así como de la matriz S se definieron en la sección II.3.

En el problema inverso, se desea estimar la función $m(x, y)$ en la región del plano

$$|x| \leq 5 \quad |y| \leq 5$$

Las expresiones para d_i , $e_i(r)$ y S_{ij} sólo dependen del kernel $K_i(r)$. Como en este problema se utiliza el mismo kernel que en el problema de validación y nos interesa determinar $m(x, y)$ en la misma región del plano, estas tres funciones se calculan con los mismos procedimientos analíticos y numéricos presentados en las secciones II.3, II.4 y II.5. En este capítulo utilizaremos un nuevo conjunto de puntos aleatorios (x_i, y_i) donde se obtienen las observaciones.

III.4 Generación del Vector de Datos

Antes de la generación del vector de datos g , se deben determinar los puntos (x_i, y_i) del plano en que se obtienen las observaciones g_i . Como la región de interés es la misma que en el problema de validación, el pseudocódigo para determinar estos puntos es el mismo que el presentado en la sección II.6. Los puntos (x_i, y_i) sobre la región de interés que resultaron del procedimiento ya descrito se muestran en la tabla VIII.

Tabla VIII: Puntos (x_i, y_i) utilizados en el problema de aplicación

i	(x_i, y_i)
1	(-5.0, 4.5)
2	(5.0, -3.5)
3	(-3.0, 0.5)
4	(1.0, 1.50)
5	(-1.0, -1.5)
6	(-2.5, 3.50)
7	(4.0, 4.0)
8	(-5.0, -3.5)
9	(3.0, -1.50)
10	(4.5, -4.0)
11	(0.0, 4.0)
12	(4.0, 0.5)
13	(-2.5, -4.0)
14	(1.5, 3.0)
15	(-2.0, -3.5)
16	(5.0, 1.5)
17	(0.5, -3.0)
18	(-3.0, 4.0)
19	(5.0, 3.5)
20	(-4.5, 3.0)

Sobre estos puntos se calculan las observaciones, resolviendo la expresión (58)

$$g_i = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x-x_i)^2} e^{-(y-y_i)^2} \left[e^{-k_1(x-x_1)^2} e^{-n_1(y-y_1)^2} + e^{-k_2(x-x_2)^2} e^{-n_2(y-y_2)^2} \right] dx dy \quad (58)$$

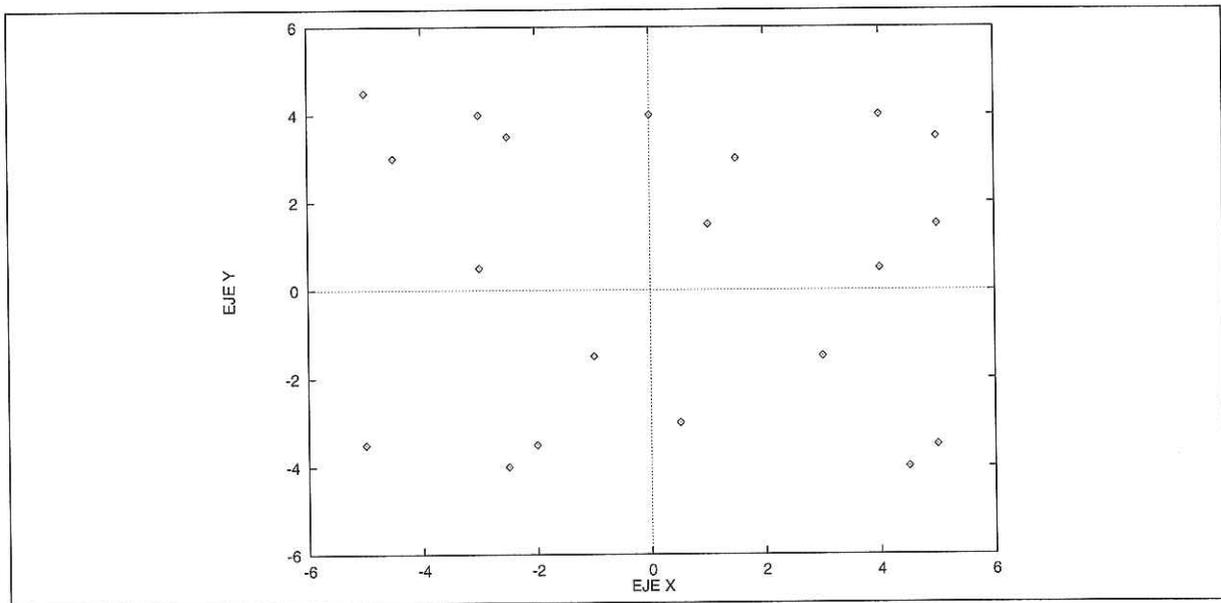


Figura 64: Gráfica de los puntos (x_i, y_i) utilizados en el problema de aplicación

Para integrar numéricamente esta expresión, se utiliza el esquema de cuatro puntos, mostrado en la sección II.4. Para aplicar este método, los límites de integración deben ser finitos. Analizando el comportamiento del kernel $K_i(x, y)$ y considerando que los puntos (x_i, y_i) se toman en la región

$$|x_i| \leq 5 \quad |y_i| \leq 5$$

la expresión a integrar numéricamente es

$$g_i = \int_{-10}^{10} \int_{-10}^{10} e^{-(x-x_i)^2} e^{-(y-y_i)^2} \left[e^{-k_1(x-x_1)^2} e^{-n_1(y-y_1)^2} + e^{-k_2(x-x_2)^2} e^{-n_2(y-y_2)^2} \right] dx dy \quad (59)$$

ya que la contribución de la integración fuera de esta región es despreciable. Las observaciones se generan con y sin ruido (gaussiano).

Primero, se generan las observaciones sin ruido, aplicando la ecuación (59) en los

puntos (x_i, y_i) . Los resultados se muestran en la tabla IX.

Tabla IX: Observaciones sin ruido para el problema de aplicación

i	(x_i, y_i)	g_i
1	(-5.0, 4.5)	0.07857534366425
2	(5.0, -3.5)	0.11196734055879
3	(-3.0, 0.5)	1.27626373180052
4	(1.0, 1.50)	2.79159386281632
5	(-1.0, -1.5)	3.43270939527530
6	(-2.5, 3.50)	0.82463330565935
7	(4.0, 4.0)	0.50629869862423
8	(-5.0, -3.5)	0.32932193921645
9	(3.0, -1.50)	1.09331116400128
10	(4.5, -4.0)	0.11858804548526
11	(0.0, 4.0)	1.50728253401604
12	(4.0, 0.5)	0.93324854581704
13	(-2.5, -4.0)	0.96422401671517
14	(1.5, 3.0)	1.99583045795111
15	(-2.0, -3.5)	1.67289637473273
16	(5.0, 1.5)	0.45050308664136
17	(0.5, -3.0)	1.58885892584916
18	(-3.0, 4.0)	0.50636849136467
19	(5.0, 3.5)	0.29640138056149
20	(-4.5, 3.0)	0.22584992064388

El siguiente paso es generar los datos contaminados con diferentes niveles de ruido.

Para la validación se consideran tres niveles de ruido: 0.2 %, 1.0 % y 5.0 %.

Para agregar ruido a los datos g_i , se utiliza la expresión

$$r = e \left[(p)(|g_i|) + (0.002)\left(\frac{\gamma}{2.0}\right) \right] \quad (60)$$

donde

e : número aleatorio con distribución gaussiana, de media cero y desviación estándar

uno.

p : valor que indica el porcentaje de ruido (0.002, 0.01, 0.05).

g_i : valor del dato en (x_i, y_i) .

γ : es el valor absoluto del valor máximo menos el valor mínimo de g en la región de interes. En este problema, $\gamma = 3.5$.

Los valores constantes utilizados en la ecuación (60) se seleccionaron de tal manera que los errores relativos fueran de magnitud similar al valor de p . Este modelo para el ruido es similar al utilizado en el capítulo anterior.

Con el primer término de la ecuación (60), se modela el error instrumental, que en general es proporcional a la magnitud de la observación, mientras que con el segundo término se esta introduciendo el error debido al modelado, evitando también que el error sea cero cuando la magnitud de la observación sea cero.

Los resultados para los diferentes niveles de ruido se muestran en la tabla X.

Tabla X: Observaciones con ruido para el problema de aplicación

i	(x_i, y_i)	g_i 0.2%	g_i 1.0%	g_i 5.0%
1	(-5.0, 4.5)	0.0845	0.0876	0.0826
2	(5.0, -3.5)	0.1099	0.1141	0.1328
3	(-3.0, 0.5)	1.2773	1.2849	1.3673
4	(1.0, 1.50)	2.8068	2.8127	2.7823
5	(-1.0, -1.5)	3.4237	3.4167	3.0974
6	(-2.5, 3.50)	0.8286	0.8036	0.8586
7	(4.0, 4.0)	0.5012	0.5075	0.5014
8	(-5.0, -3.5)	0.3272	0.3287	0.3371
9	(3.0, -1.50)	1.0877	1.0996	1.0578
10	(4.5, -4.0)	0.1184	0.1217	0.1054
11	(0.0, 4.0)	1.5202	1.5027	1.6613
12	(4.0, 0.5)	0.9319	0.9346	0.9149
13	(-2.5, -4.0)	0.9626	0.9596	1.0109
14	(1.5, 3.0)	1.9930	2.0321	1.9247
15	(-2.0, -3.5)	1.6707	1.6638	1.7225
16	(5.0, 1.5)	0.4478	0.4543	0.4089
17	(0.5, -3.0)	1.6016	1.6121	1.7168
18	(-3.0, 4.0)	0.5132	0.5071	0.4527
19	(5.0, 3.5)	0.2954	0.3074	0.2904
20	(-4.5, 3.0)	0.2259	0.2239	0.2340

III.5 Solución del Problema Inverso

Para el caso de observaciones sin ruido, la estimación $\hat{m}(r)$, el error de la estimación y los residuales, para $\alpha = 1.0$, se muestran en las figuras 65 a la 67.

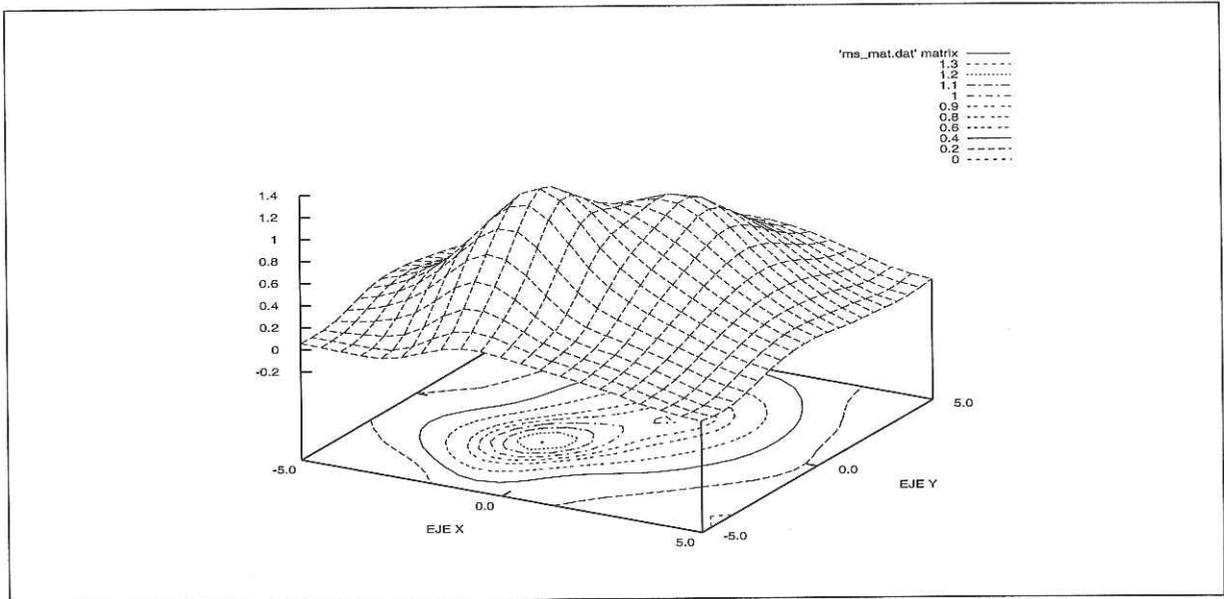


Figura 65: Gráfica de $\hat{m}(r)$ para datos sin ruido

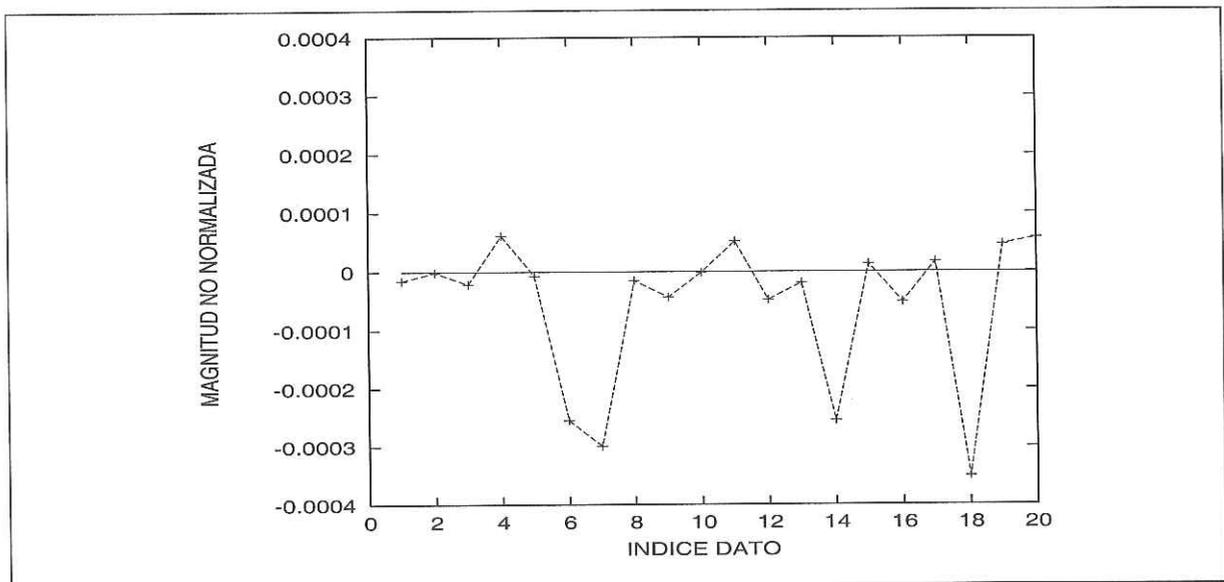


Figura 66: Gráfica de residuales para datos sin ruido

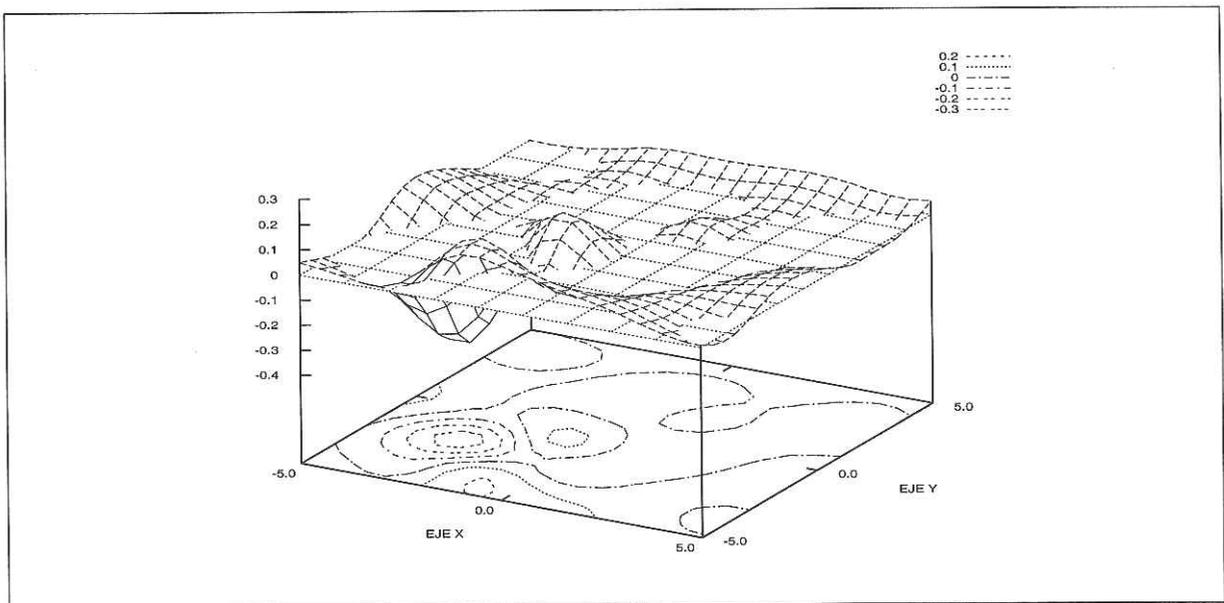


Figura 67: Gráfica del error para datos sin ruido

Para los casos de observaciones con ruido, la estimación $\hat{m}(r)$, el error de la estimación y los residuales normalizados se muestran a continuación.

Los resultados para un 5.0% de error se muestran en las figuras 68 a la 85.

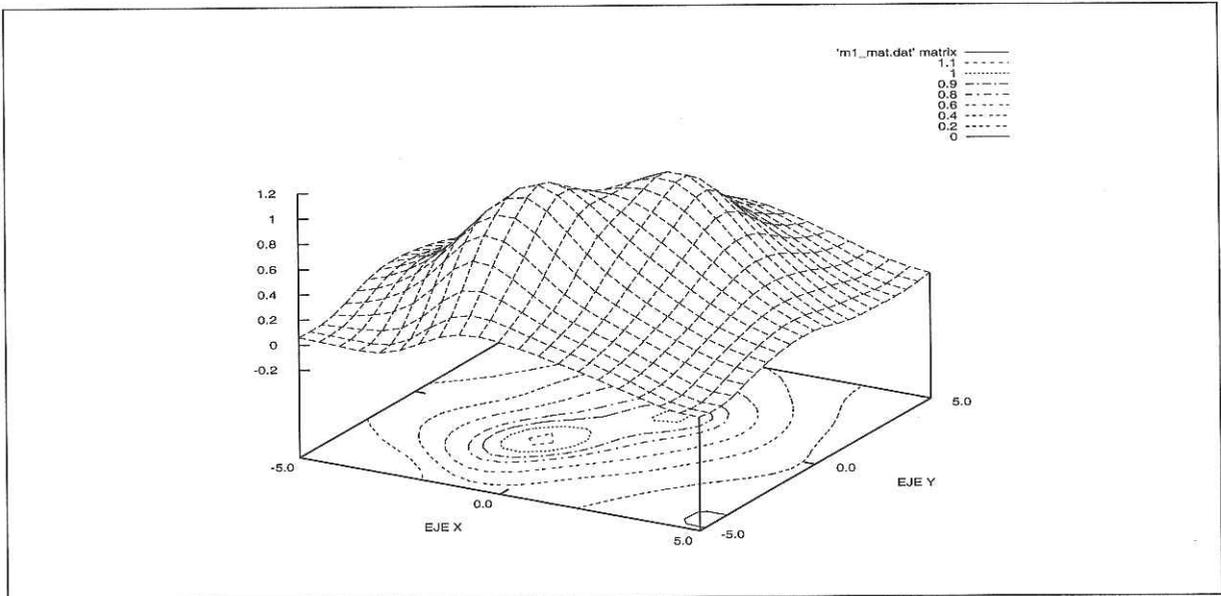


Figura 68: Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 5.0% de error

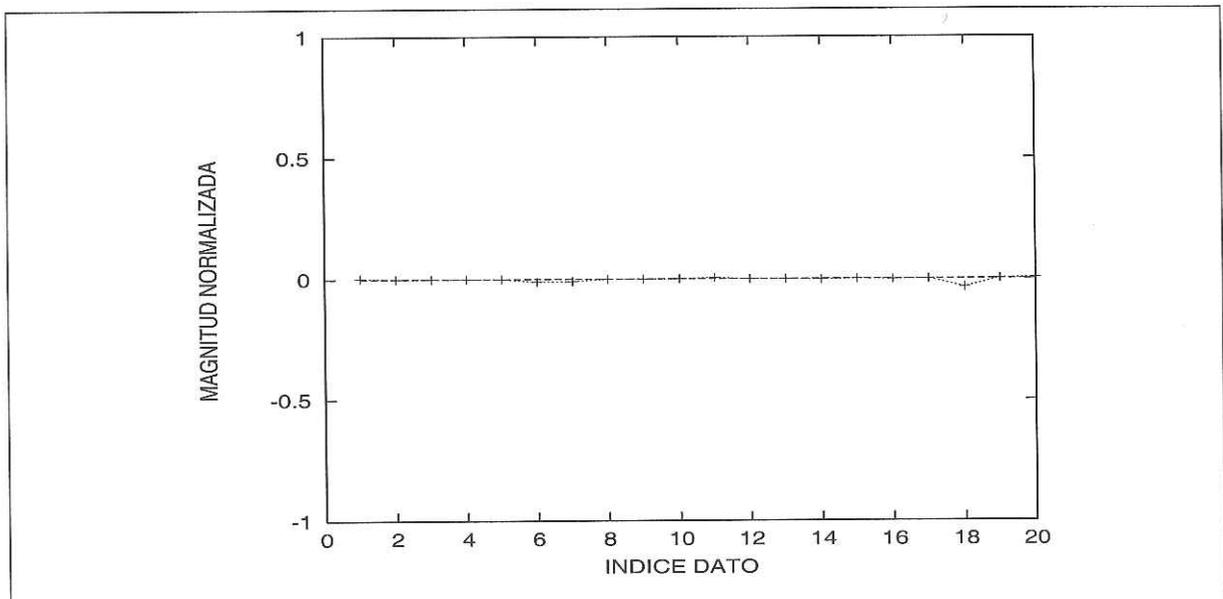


Figura 69: Gráfica de residuales para $\alpha = 0.00001$; 5.0% de error

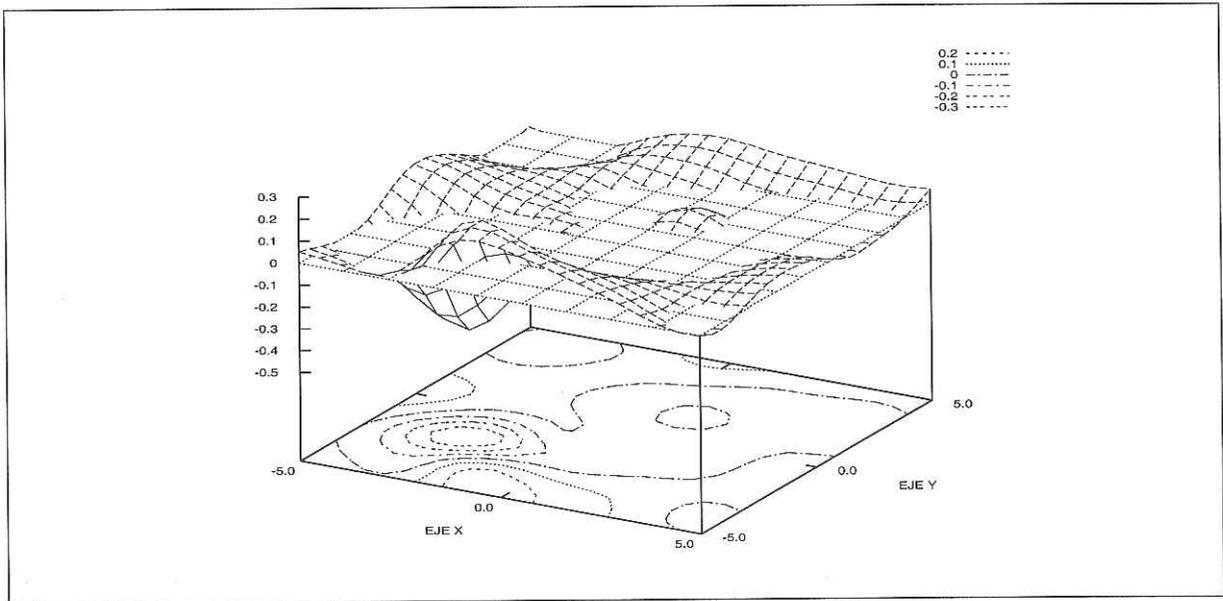


Figura 70: Gráfica del error para $\alpha = 0.00001$; 5.0% de error

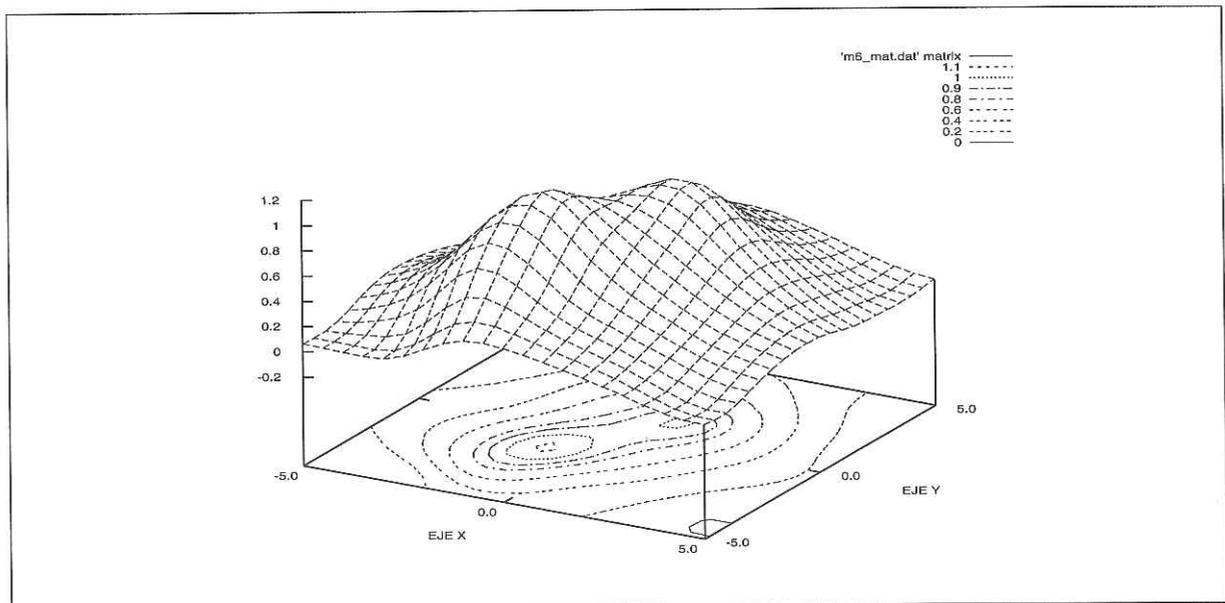


Figura 71: Gráfica de $\hat{m}(r)$ para $\alpha = 1.0$; 5.0% de error

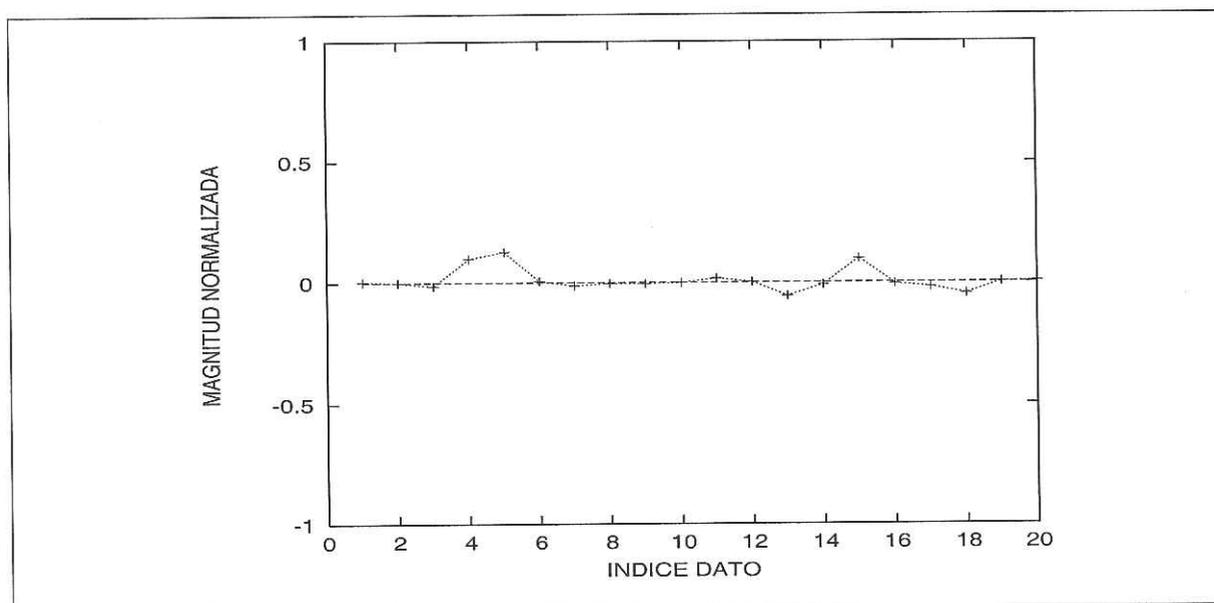


Figura 72: Gráfica de residuales para $\alpha = 1.0$; 5.0% de error

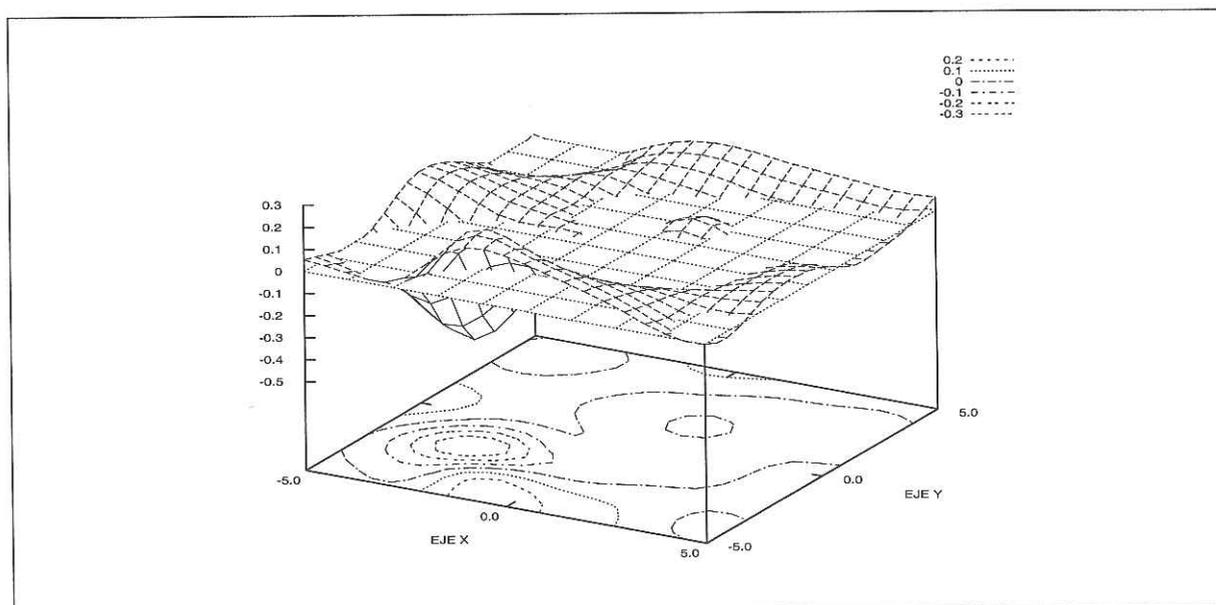


Figura 73: Gráfica del error para $\alpha = 1.0$; 5.0% de error

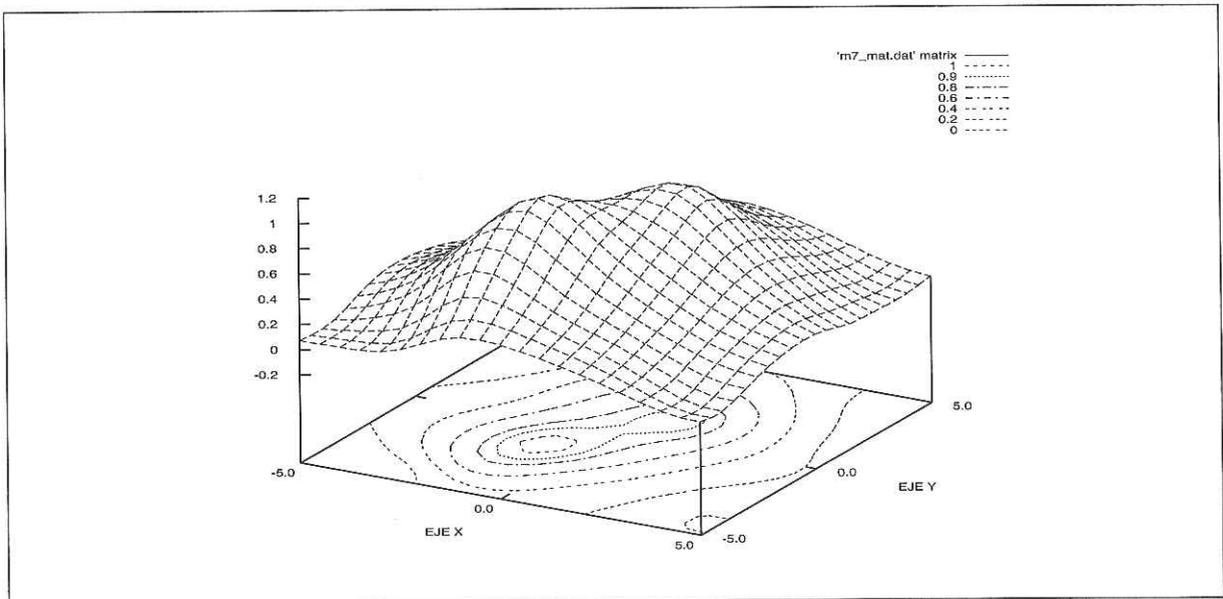


Figura 74: Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 5.0% de error

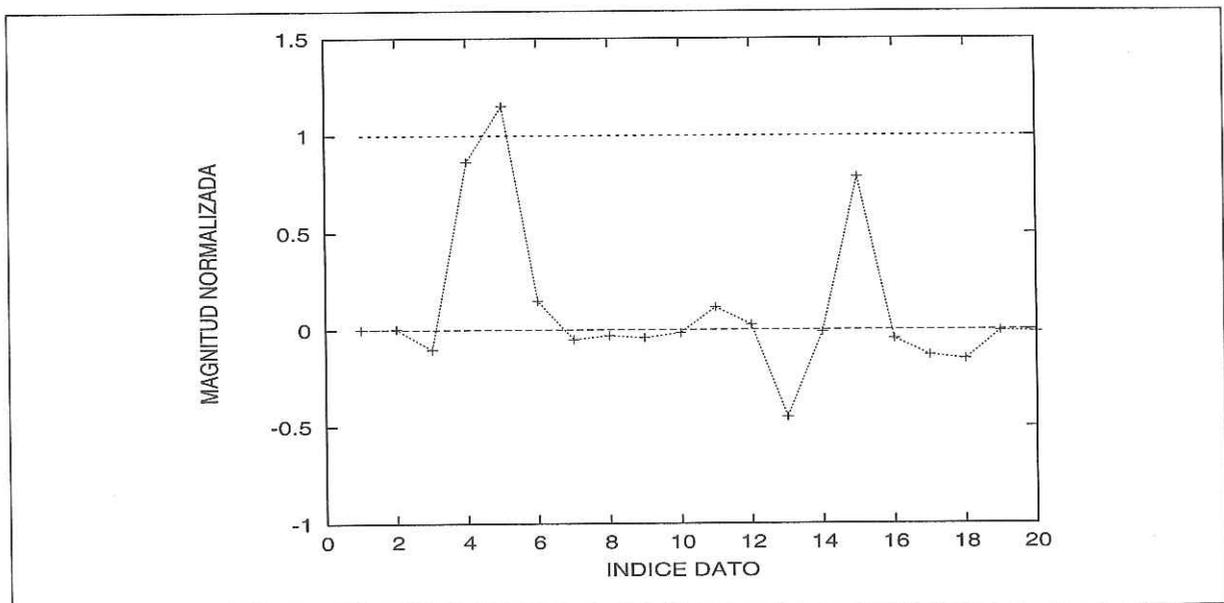


Figura 75: Gráfica de residuales para $\alpha = 10.0$; 5.0% de error

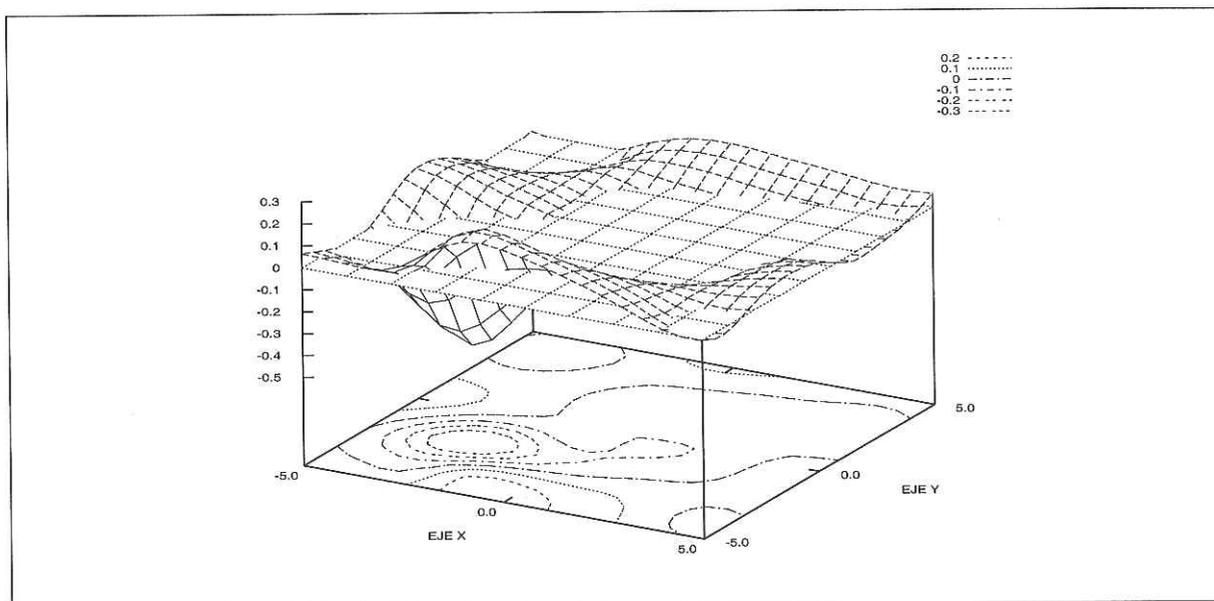


Figura 76: Gráfica del error para $\alpha = 10.0$; 5.0% de error

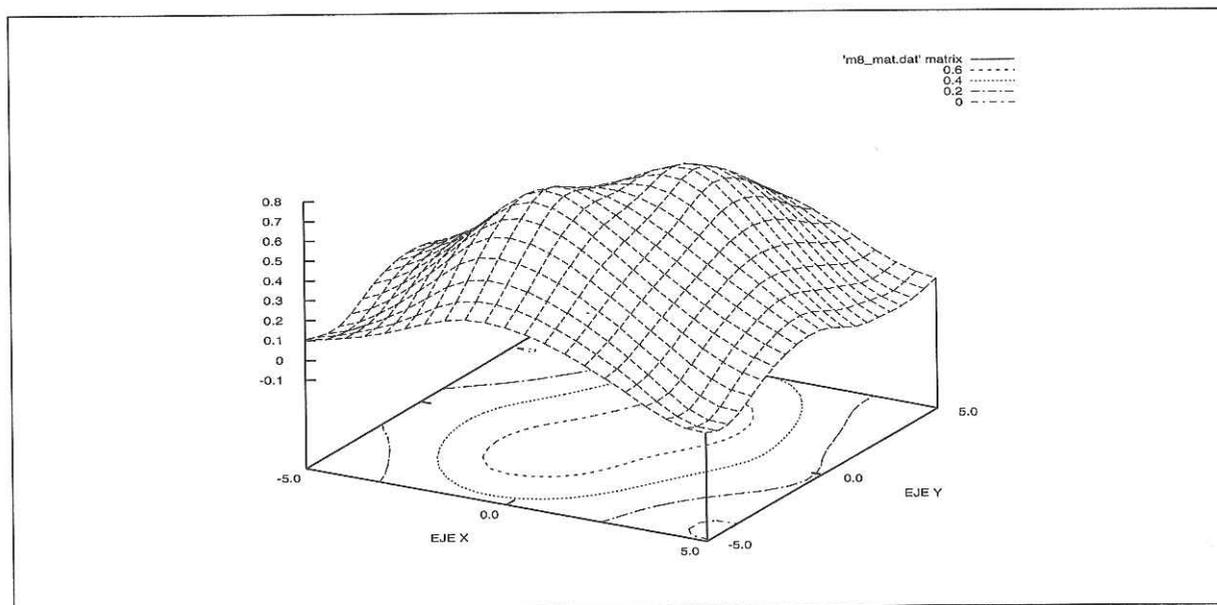


Figura 77: Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 5.0% de error

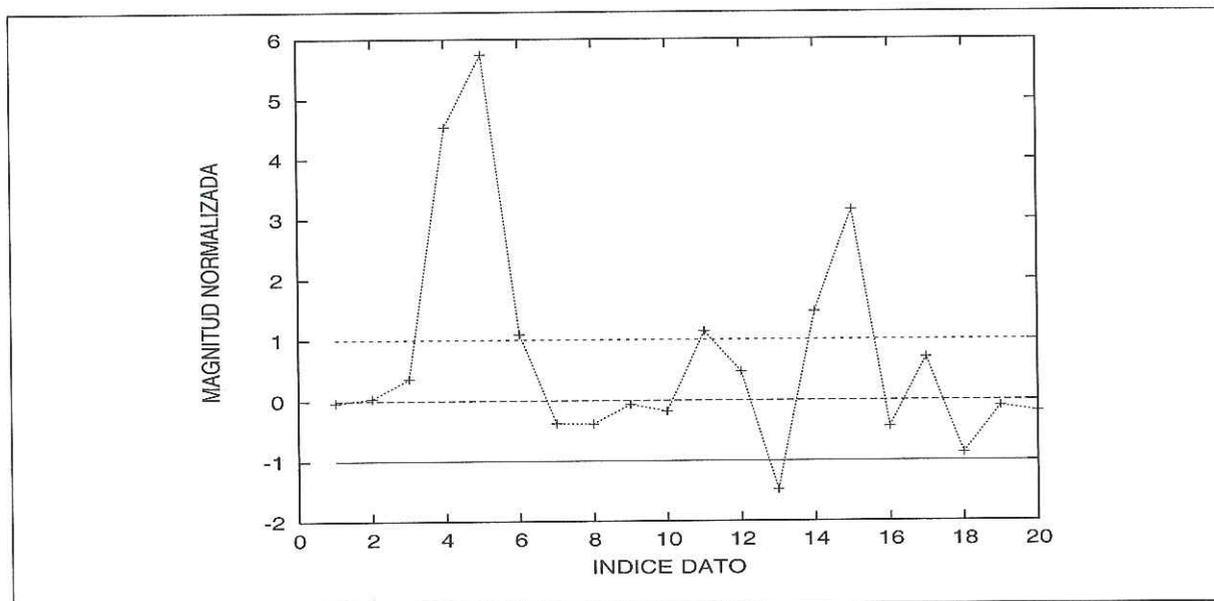


Figura 78: Gráfica de residuales para $\alpha = 100.0$; 5.0% de error

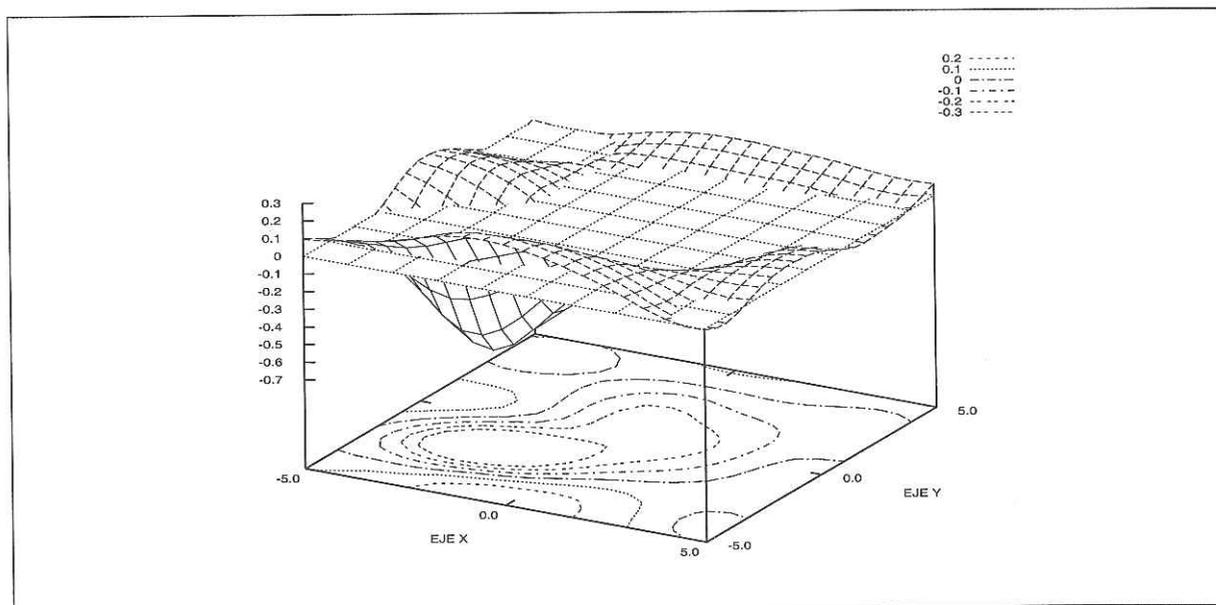


Figura 79: Gráfica del error para $\alpha = 100.0$; 5.0% de error

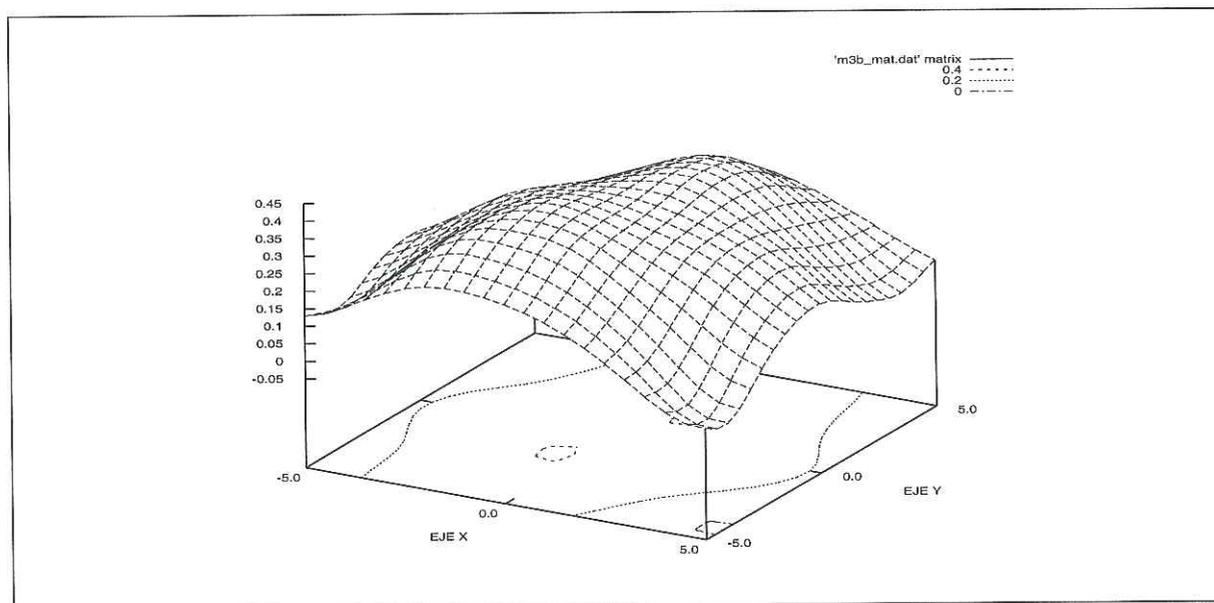


Figura 80: Gráfica de $\hat{m}(r)$ para $\alpha = 1000.0$; 5.0% de error

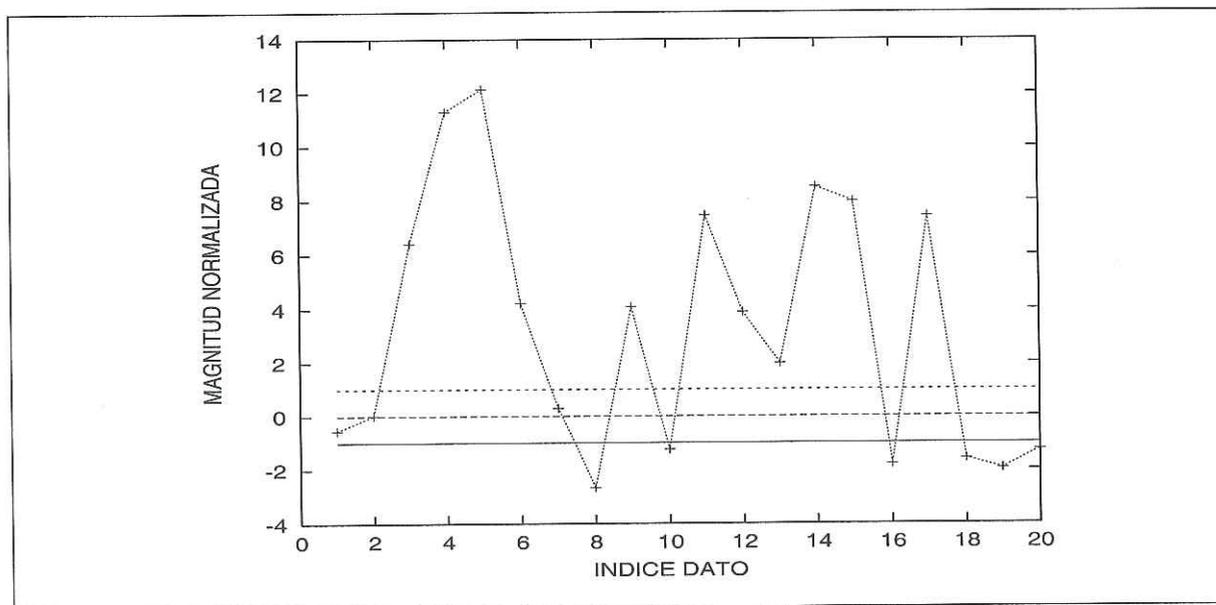


Figura 81: Gráfica de residuales para $\alpha = 1000.0$; 5.0% de error

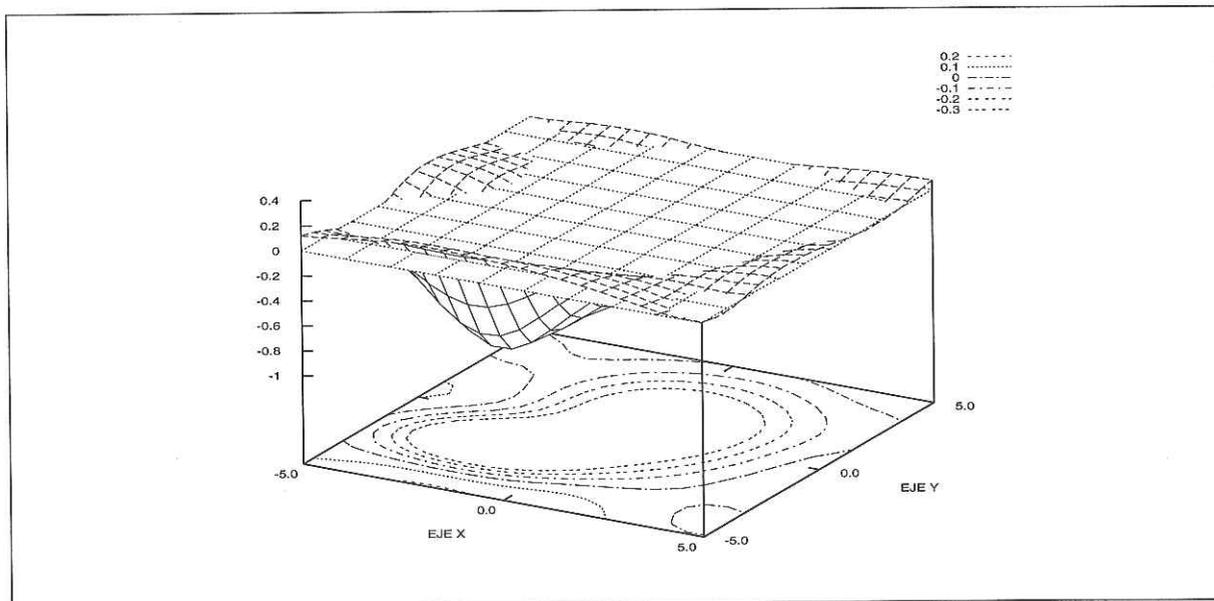


Figura 82: Gráfica del error para $\alpha = 1000.0$; 5.0% de error

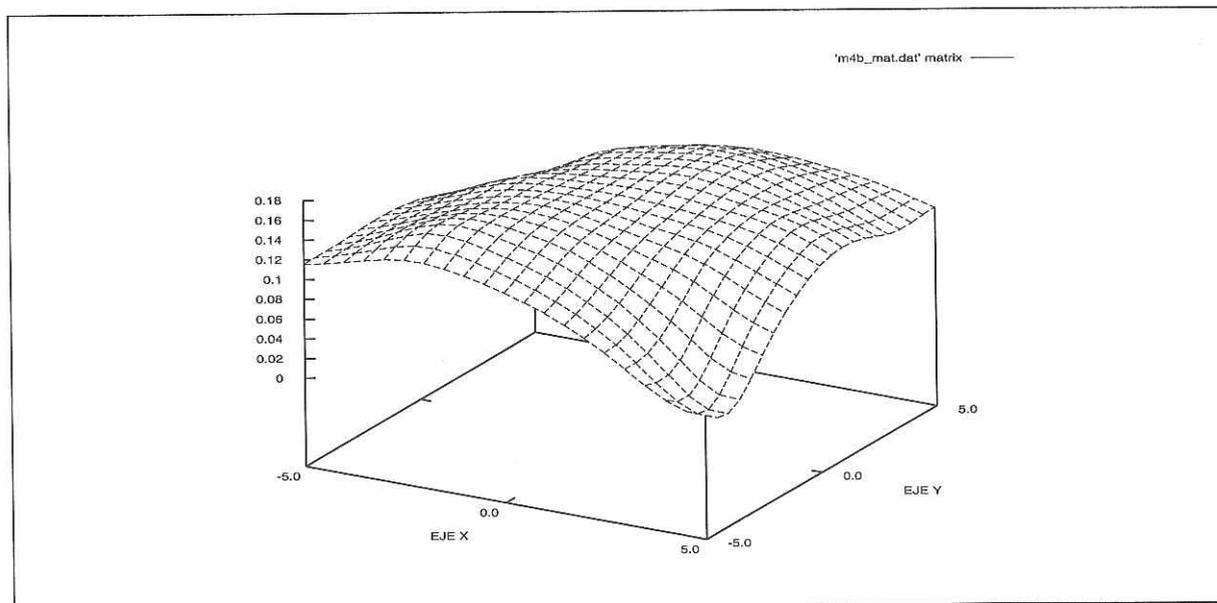


Figura 83: Gráfica de $\hat{m}(r)$ para $\alpha = 10000.0$; 5.0% de error

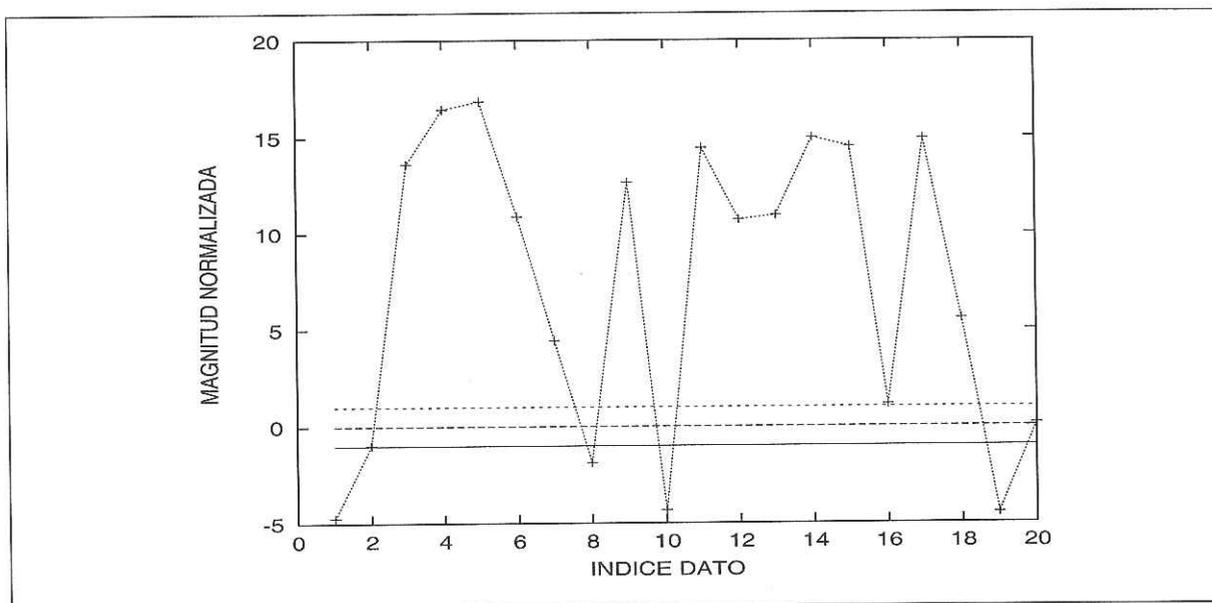


Figura 84: Gráfica de residuales para $\alpha = 10000.0$; 5.0% de error

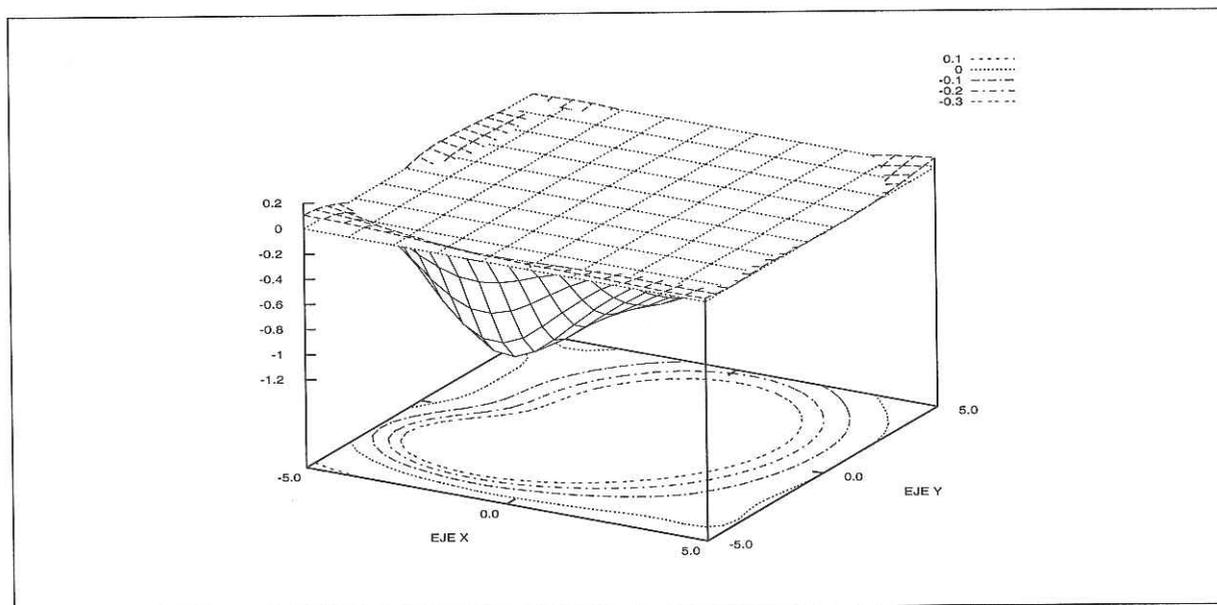


Figura 85: Gráfica del error para $\alpha = 10000.0$; 5.0% de error

Los resultados para un 1.0% de error se muestran en las figuras 86 a la 100.

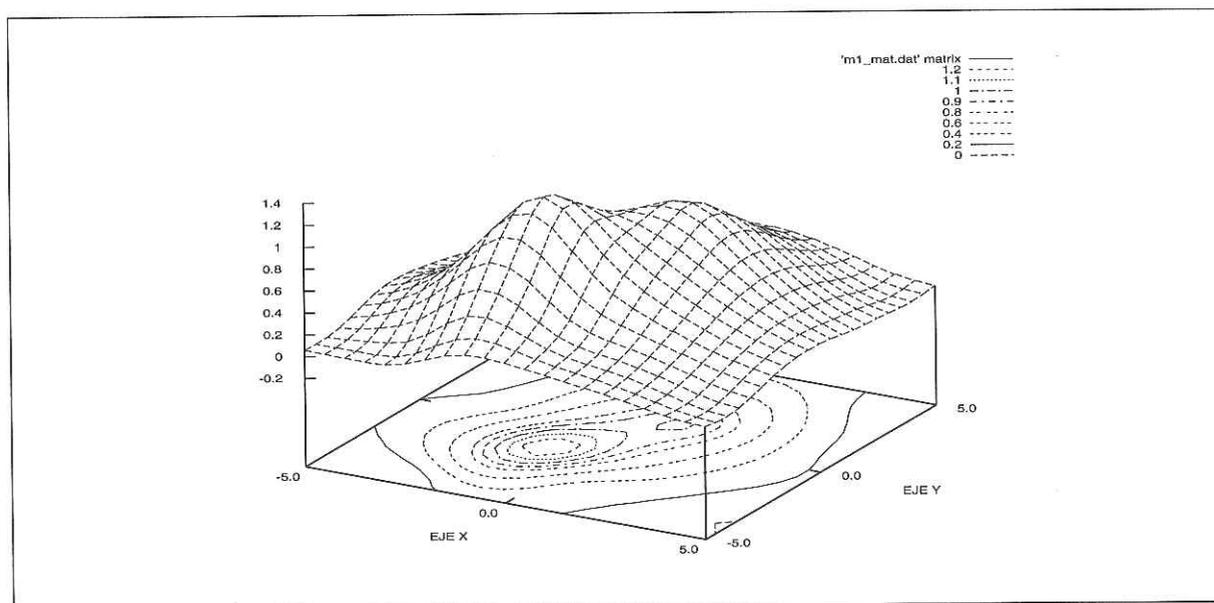


Figura 86: Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 1.0% de error

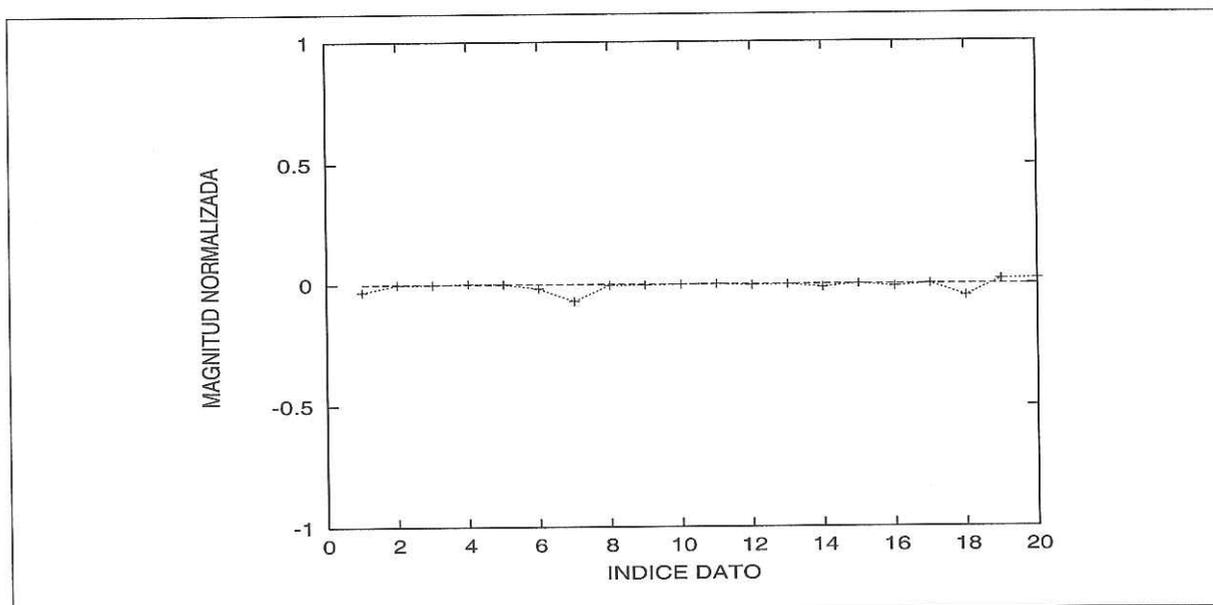


Figura 87: Gráfica de residuales para $\alpha = 0.00001$; 1.0% de error

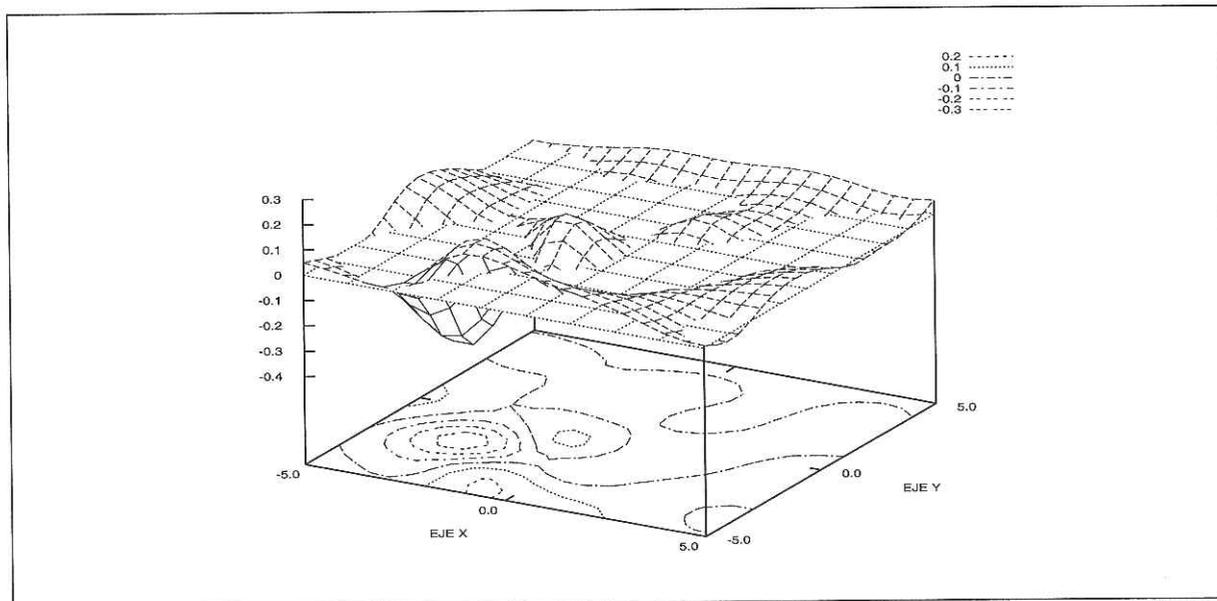


Figura 88: Gráfica del error para $\alpha = 0.00001$; 1.0% de error

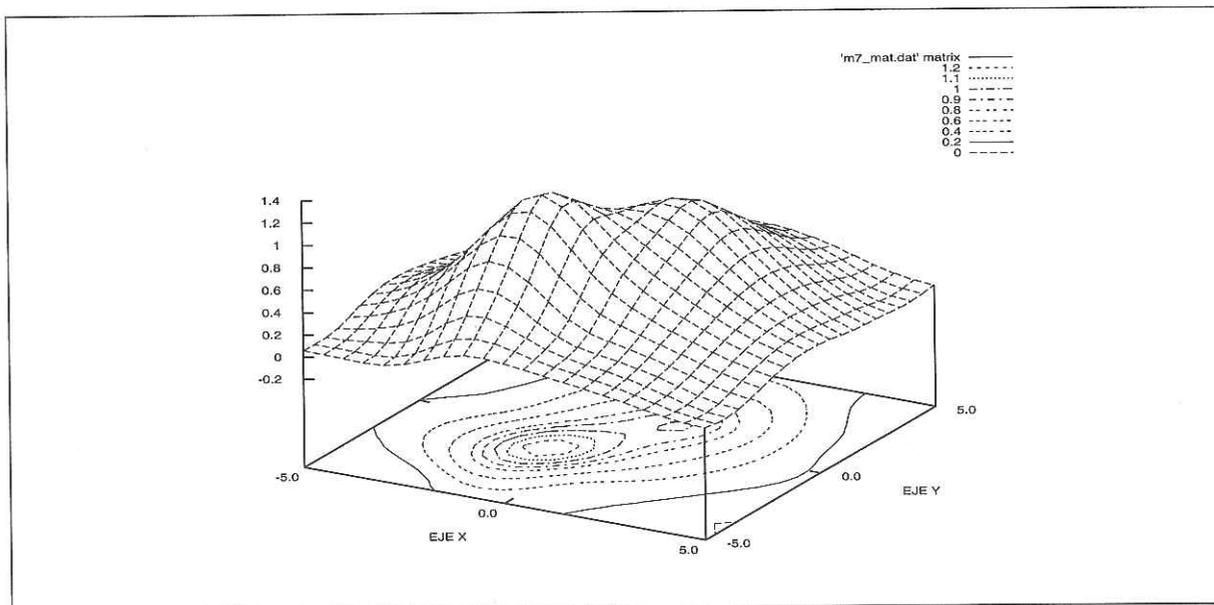


Figura 89: Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 1.0% de error

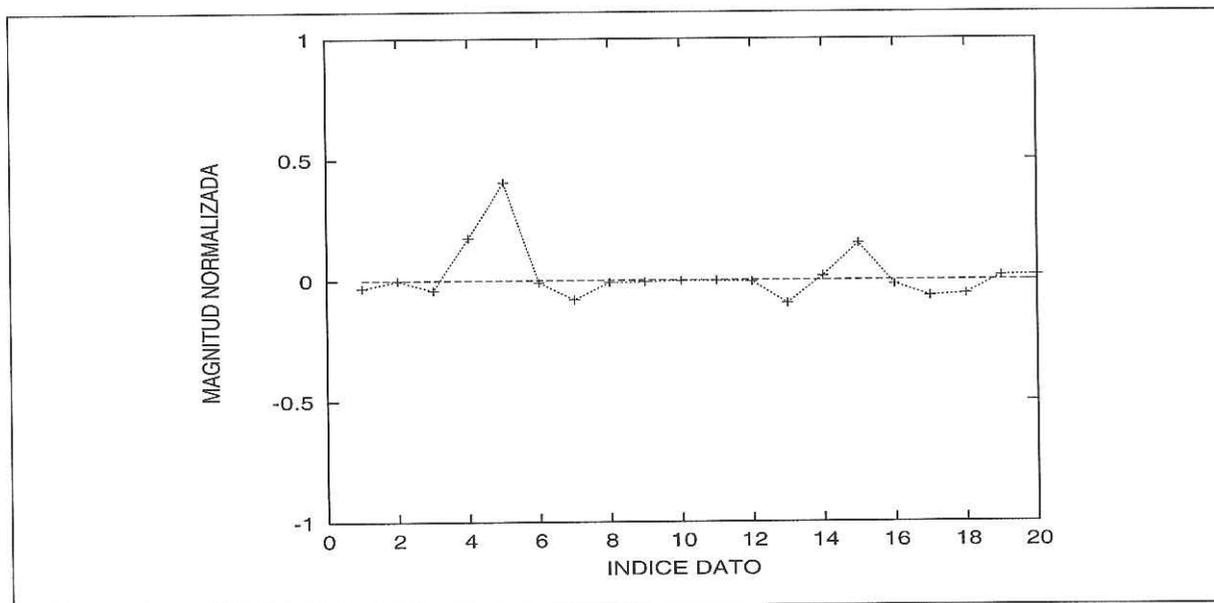


Figura 90: Gráfica de residuales para $\alpha = 10.0$; 1.0% de error

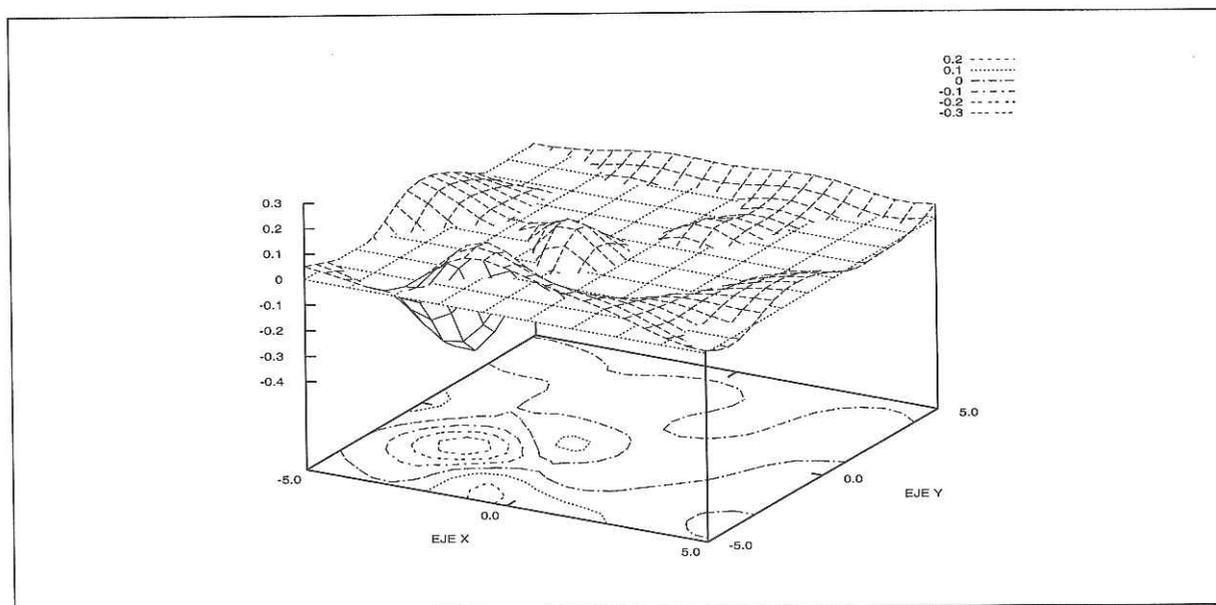


Figura 91: Gráfica del error para $\alpha = 10.0$; 1.0% de error

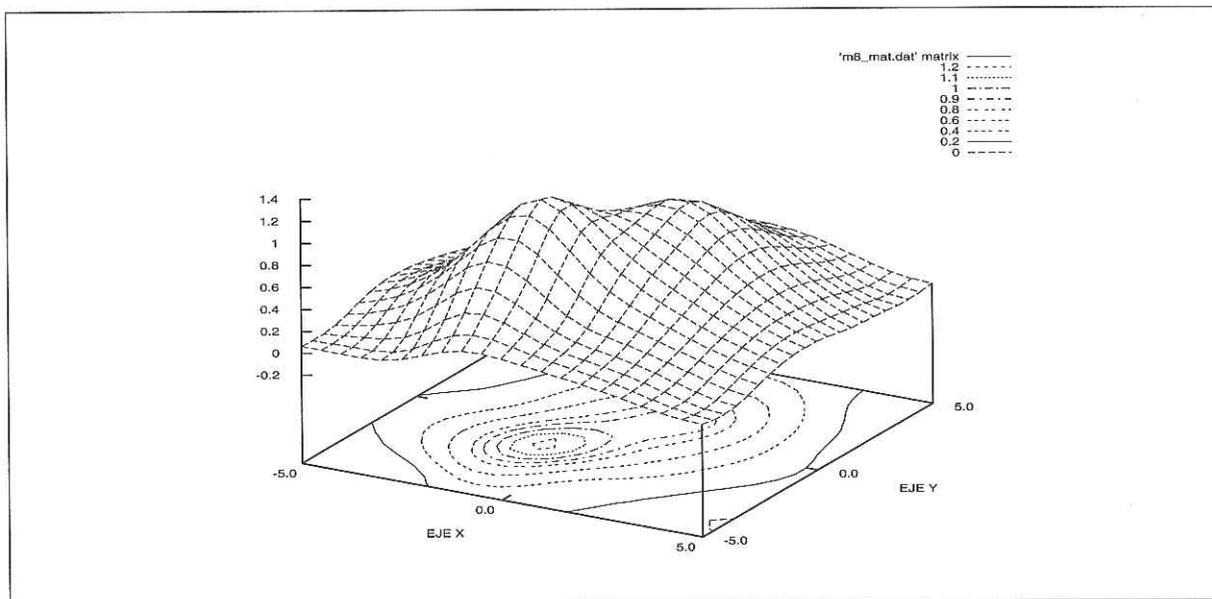


Figura 92: Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 1.0% de error

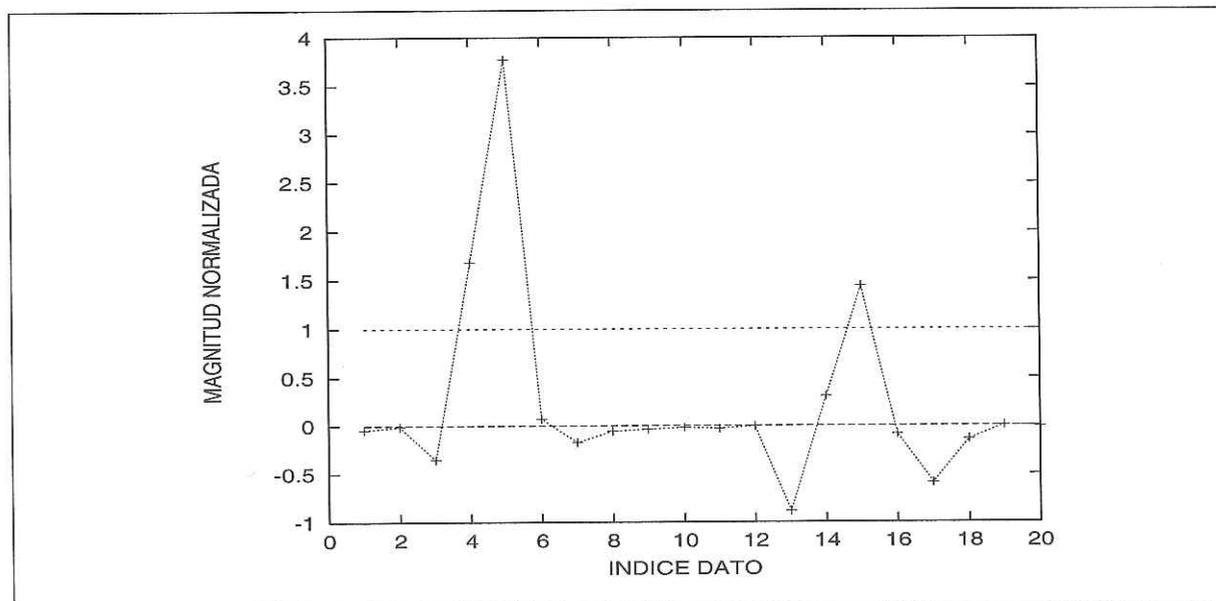


Figura 93: Gráfica de residuales para $\alpha = 100.0$; 1.0% de error

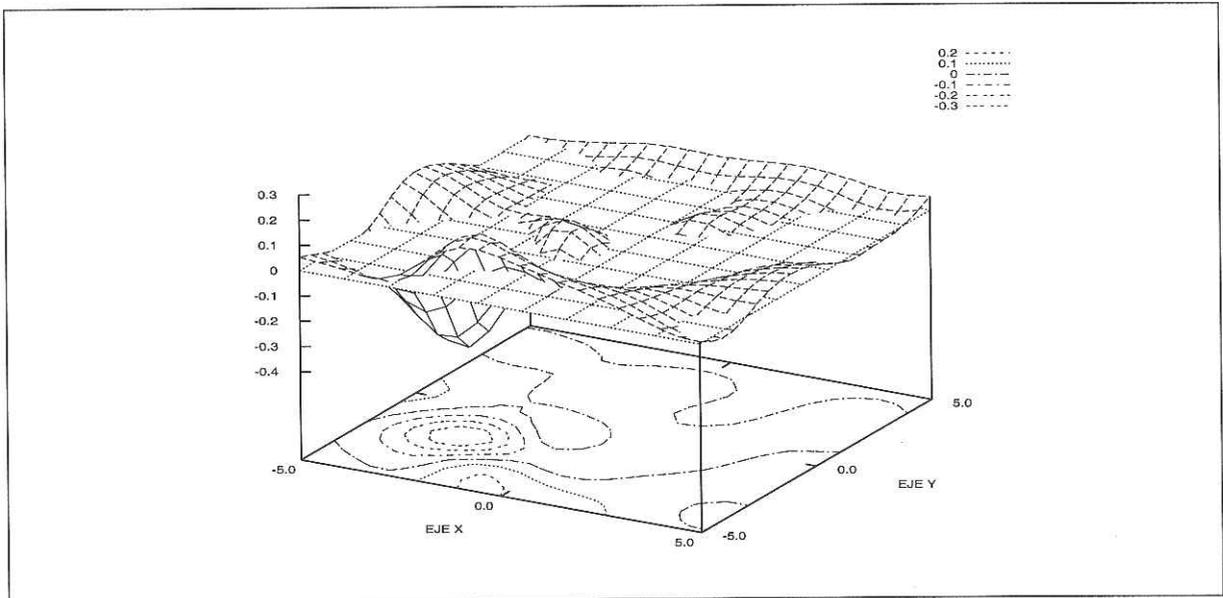


Figura 94: Gráfica del error para $\alpha = 100.0$; 1.0% de error

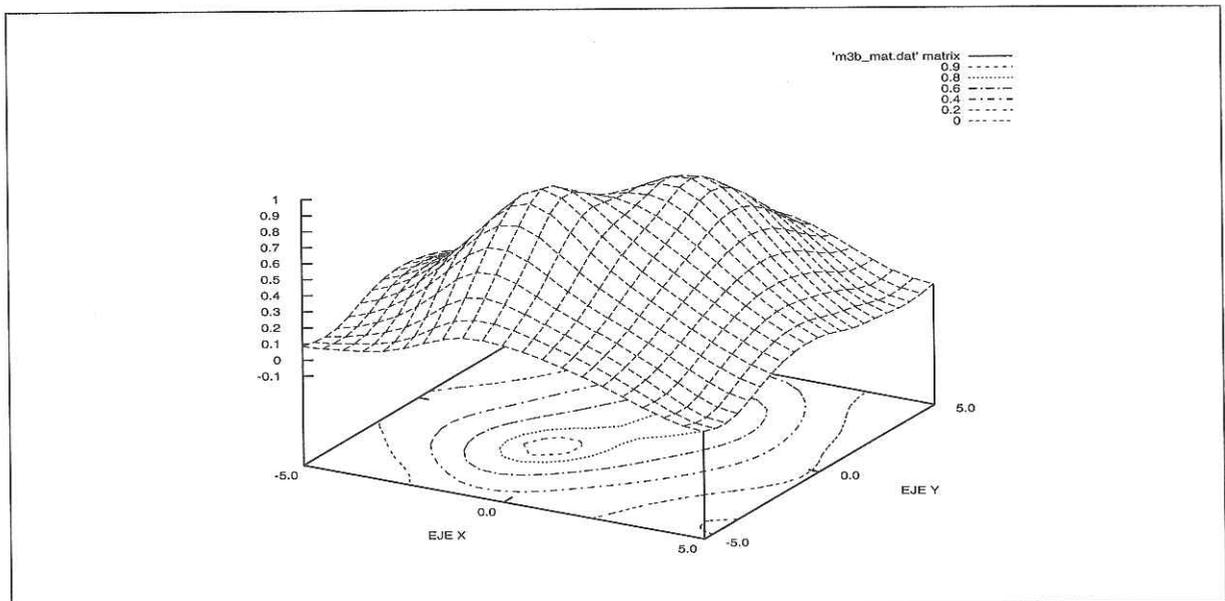


Figura 95: Gráfica de $\hat{m}(r)$ para $\alpha = 1000.0$; 1.0% de error

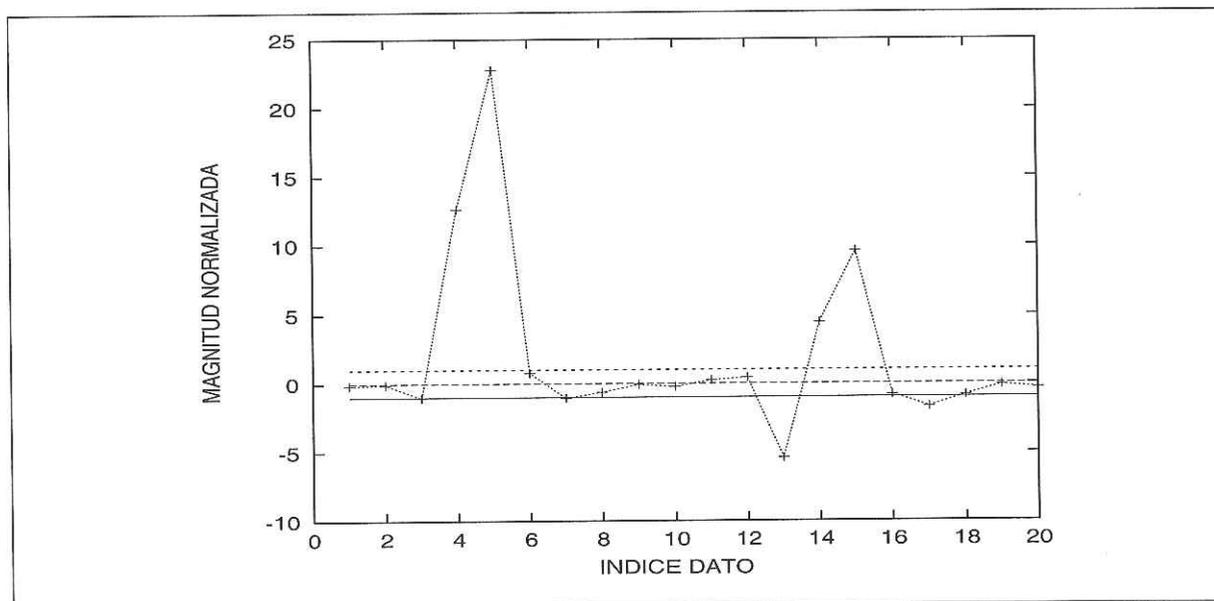


Figura 96: Gráfica de residuales para $\alpha = 1000.0$; 1.0% de error

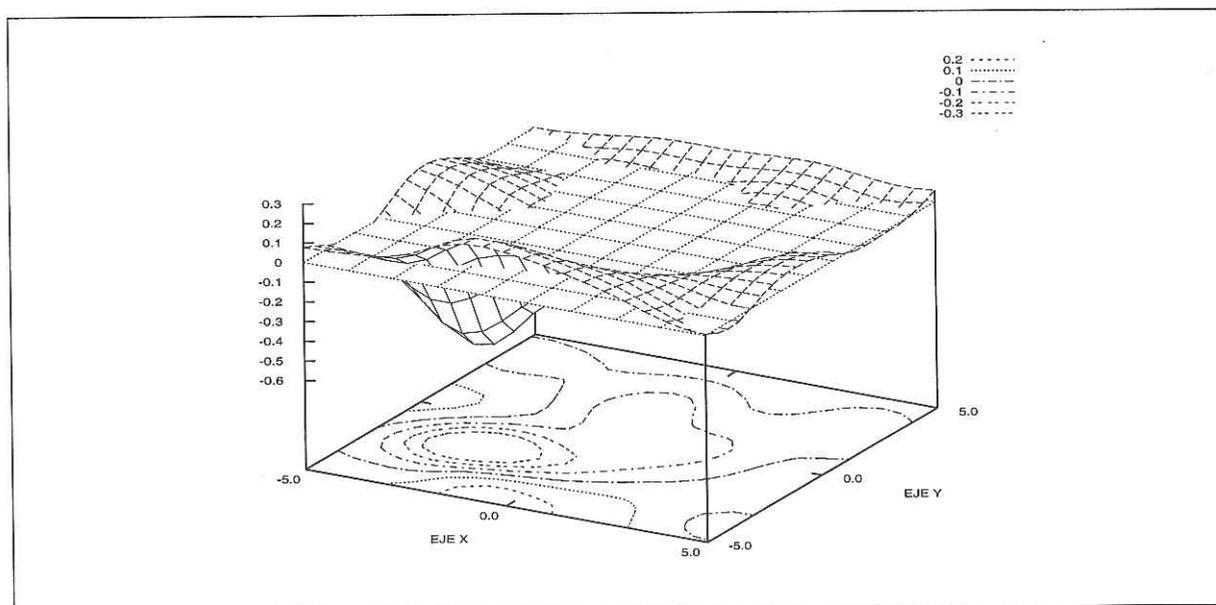


Figura 97: Gráfica del error para $\alpha = 1000.0$; 1.0% de error

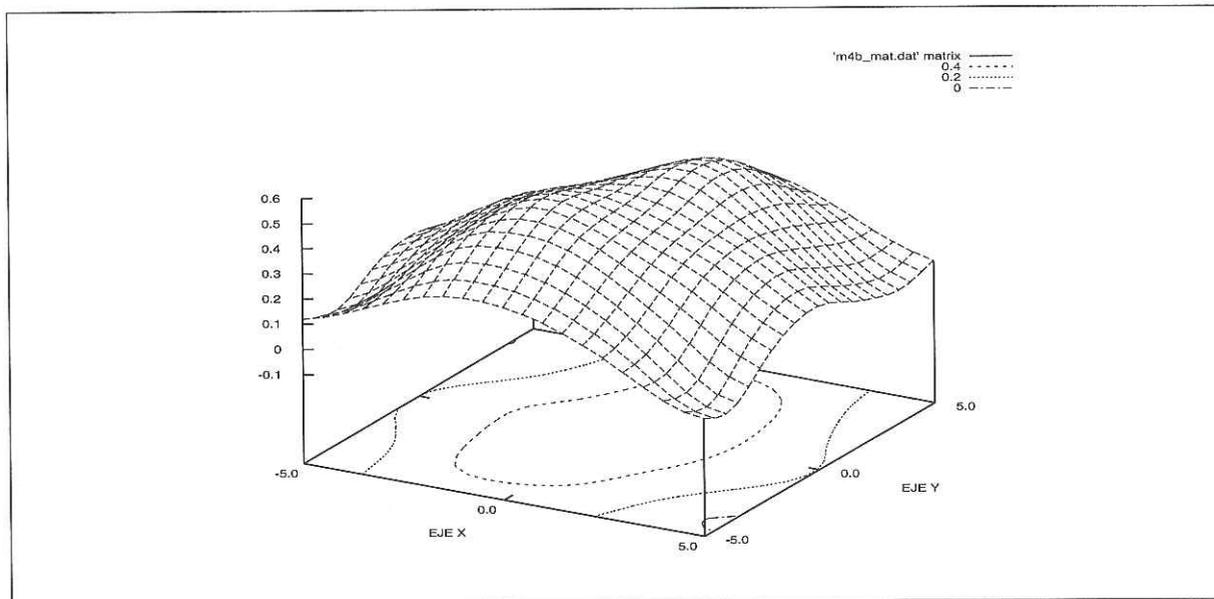


Figura 98: Gráfica de $\hat{m}(r)$ para $\alpha = 10000.0$; 1.0% de error

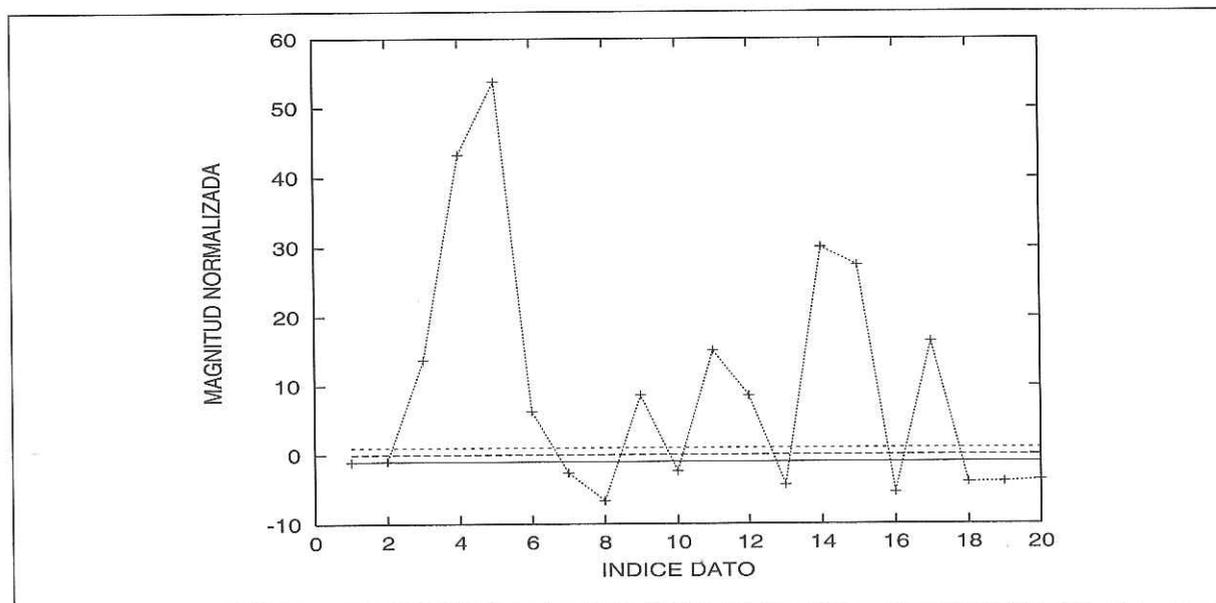


Figura 99: Gráfica de residuales para $\alpha = 10000.0$; 1.0% de error

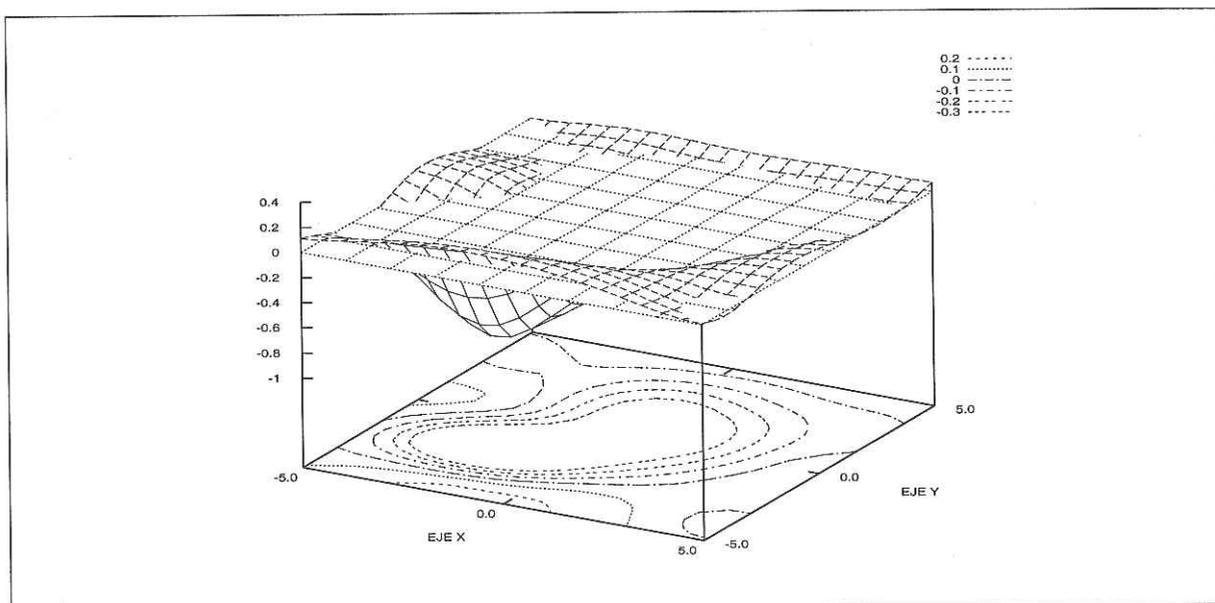


Figura 100: Gráfica del error para $\alpha = 10000.0$; 1.0% de error

Los resultados para un 0.2% de error se muestran en las figuras 101 a la 115.

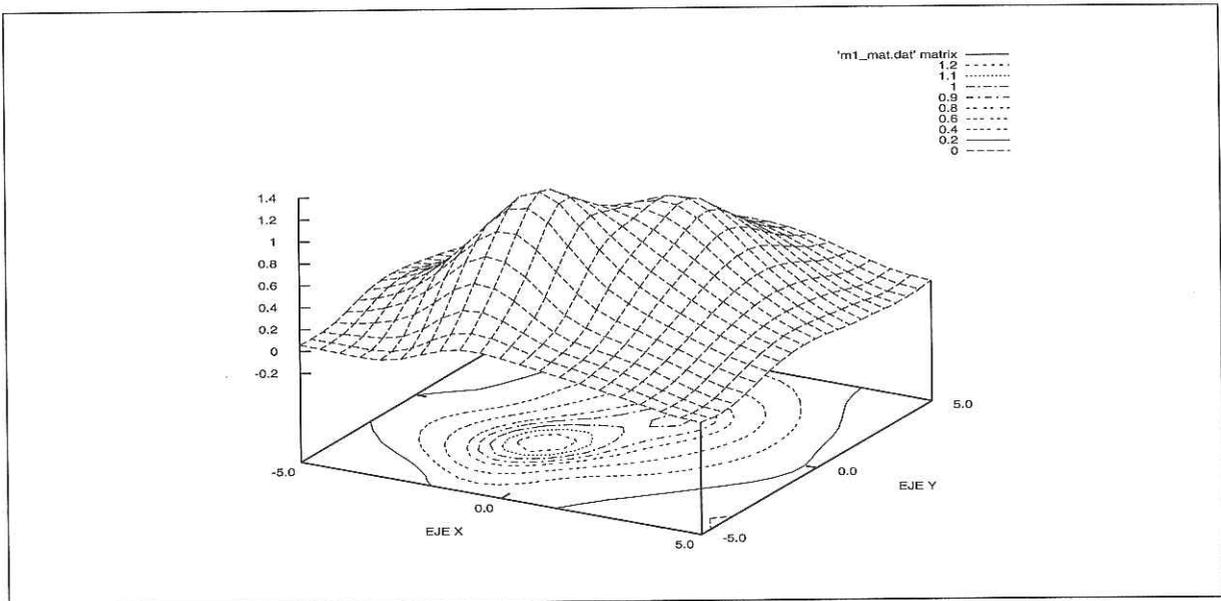


Figura 101: Gráfica de $\hat{m}(r)$ para $\alpha = 0.00001$; 0.2% de error

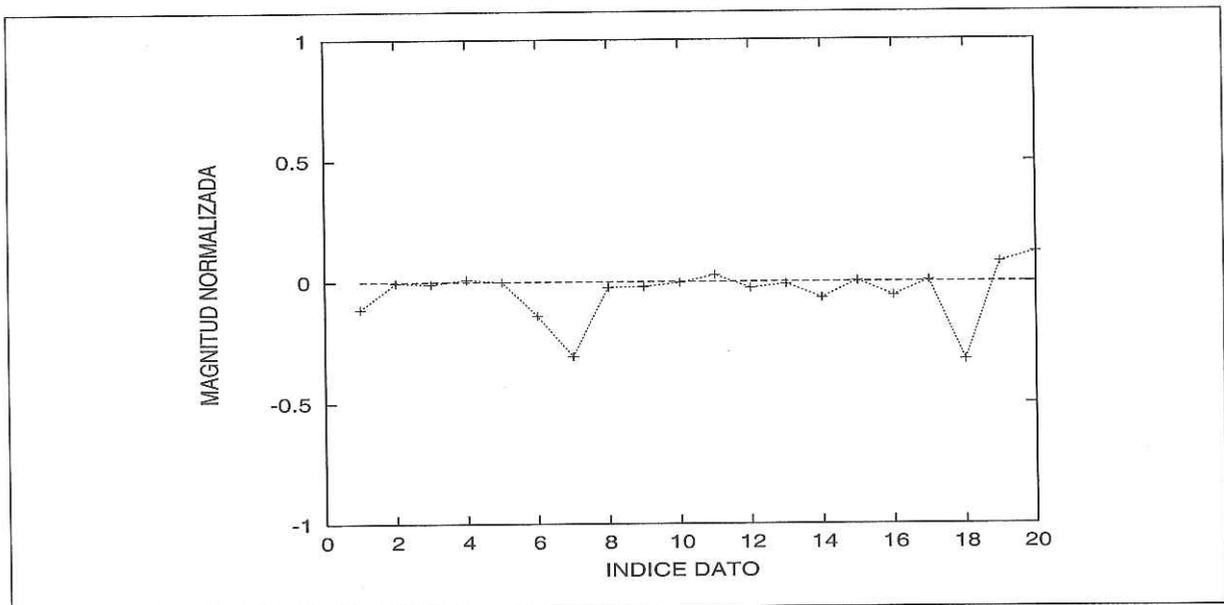


Figura 102: Gráfica de residuales para $\alpha = 0.00001$; 0.2% de error

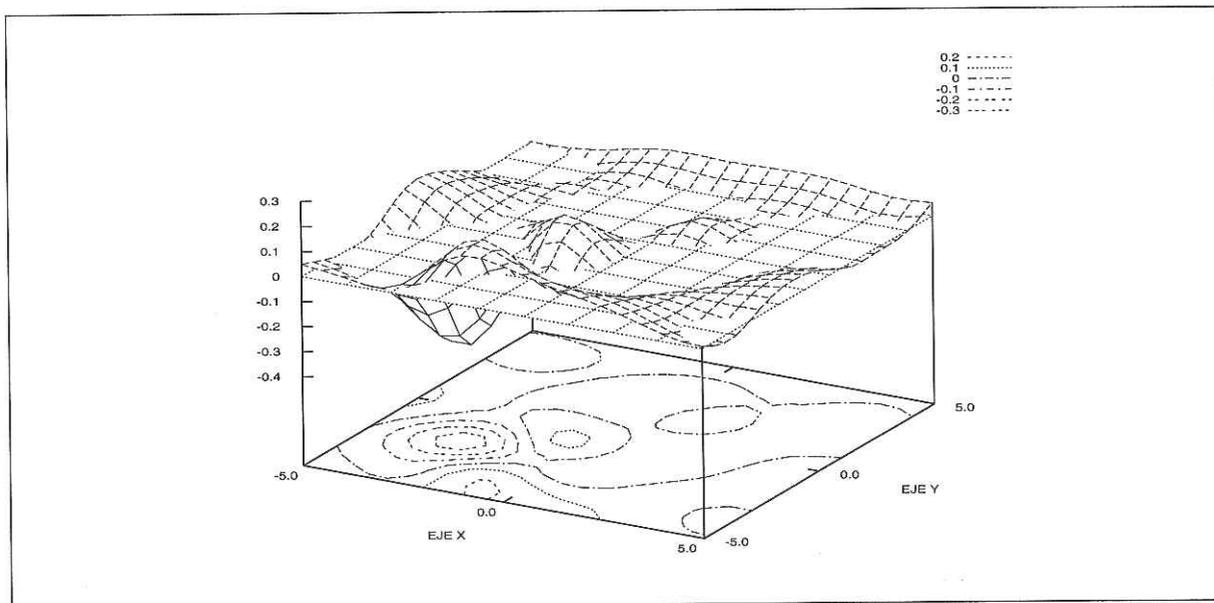


Figura 103: Gráfica del error para $\alpha = 0.00001$; 0.2% de error

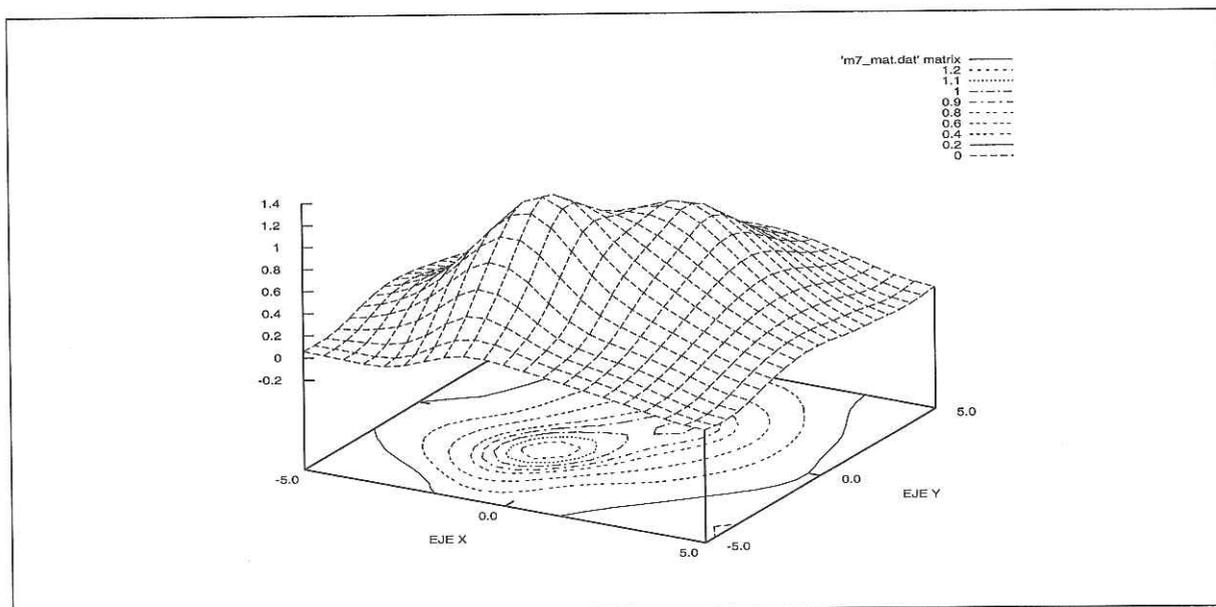


Figura 104: Gráfica de $\hat{m}(r)$ para $\alpha = 10.0$; 0.2% de error

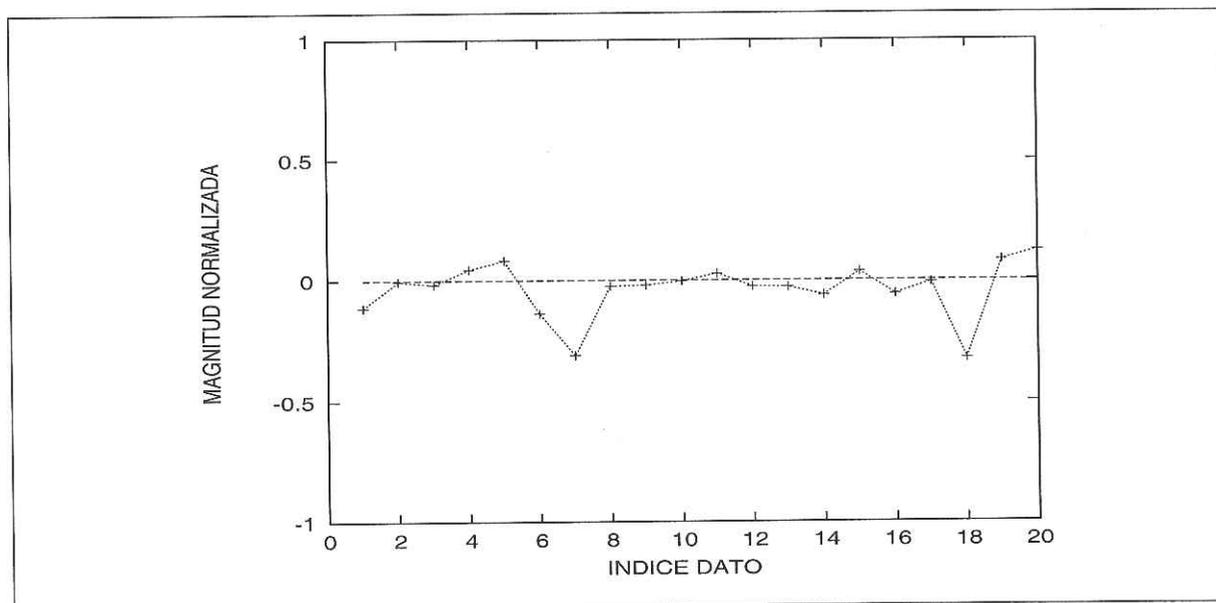


Figura 105: Gráfica de residuales para $\alpha = 10.0$; 0.2% de error

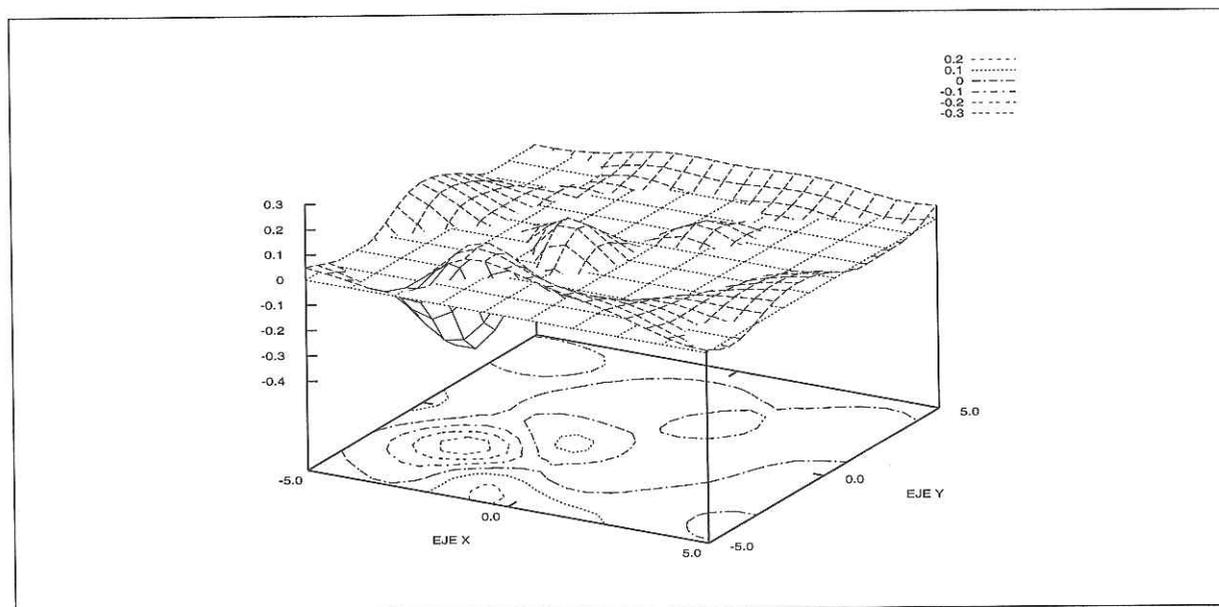


Figura 106: Gráfica del error para $\alpha = 10.0$; 0.2% de error

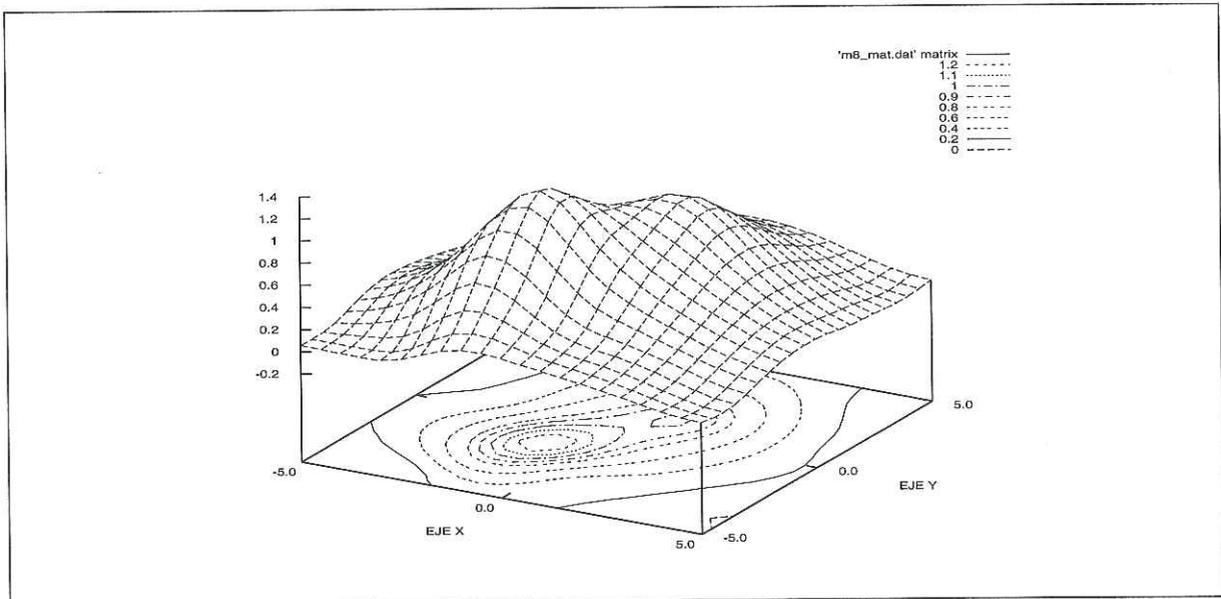


Figura 107: Gráfica de $\hat{m}(r)$ para $\alpha = 100.0$; 0.2% de error

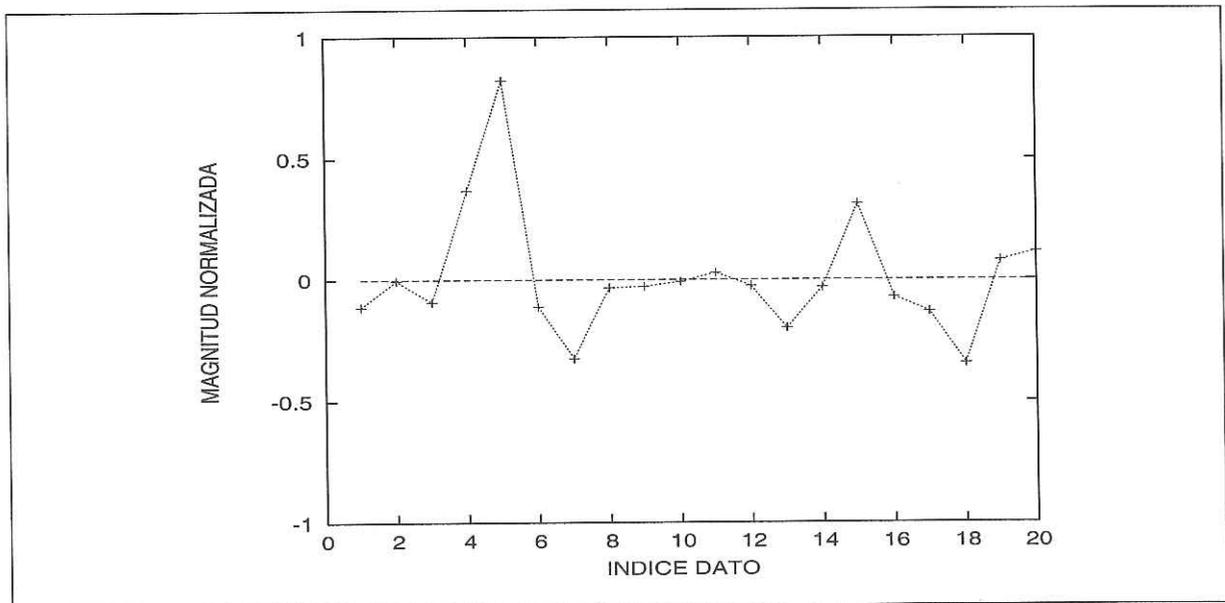


Figura 108: Gráfica de residuales para $\alpha = 100.0$; 0.2% de error

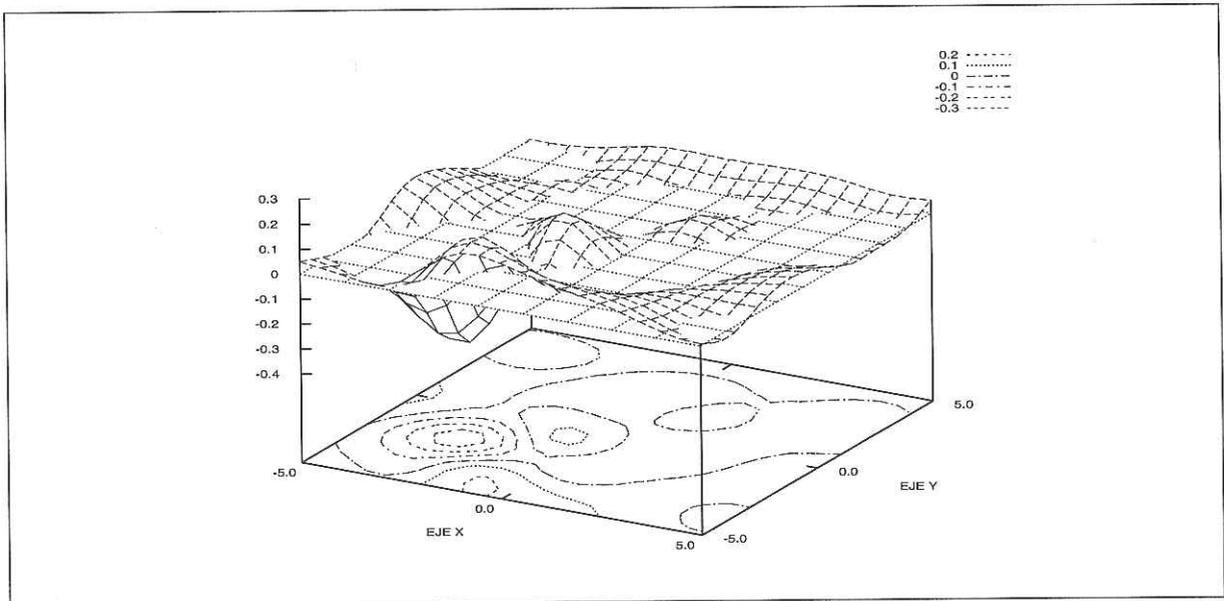


Figura 109: Gráfica del error para $\alpha = 100.0$; 0.2% de error

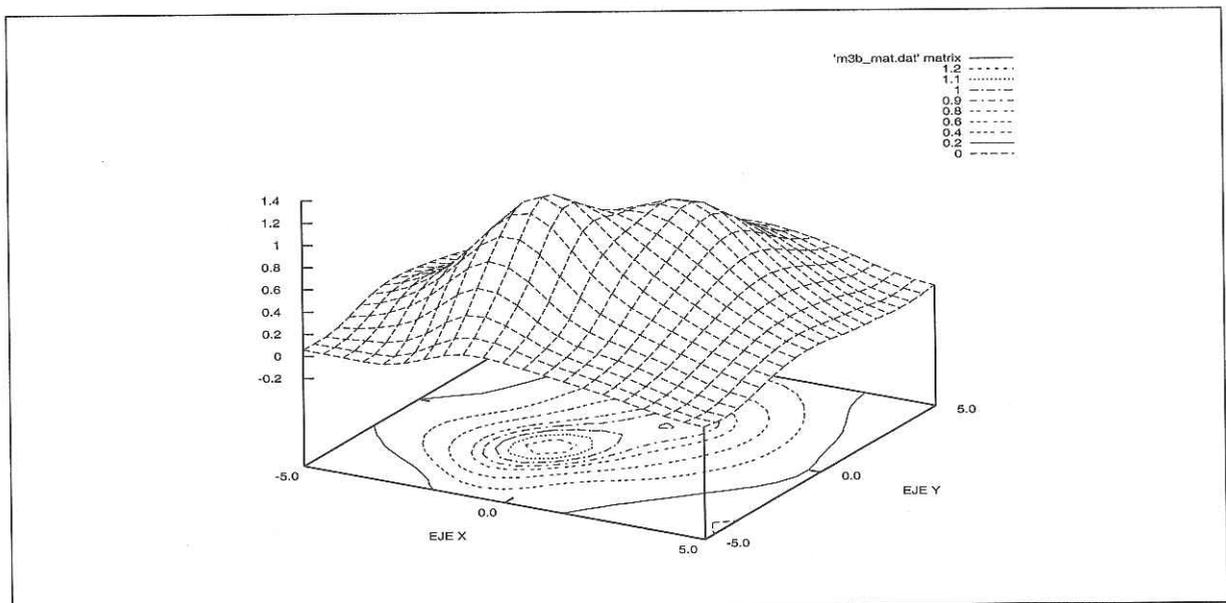


Figura 110: Gráfica de $\hat{m}(r)$ para $\alpha = 1000.0$; 0.2% de error

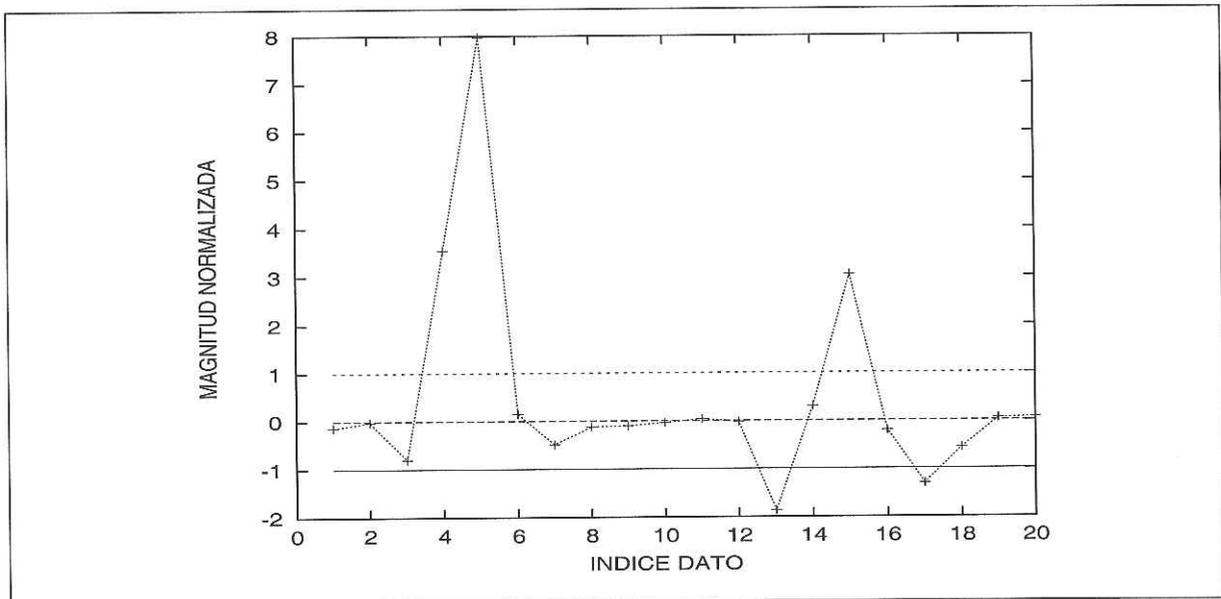


Figura 111: Gráfica de residuales para $\alpha = 1000.0$; 0.2% de error

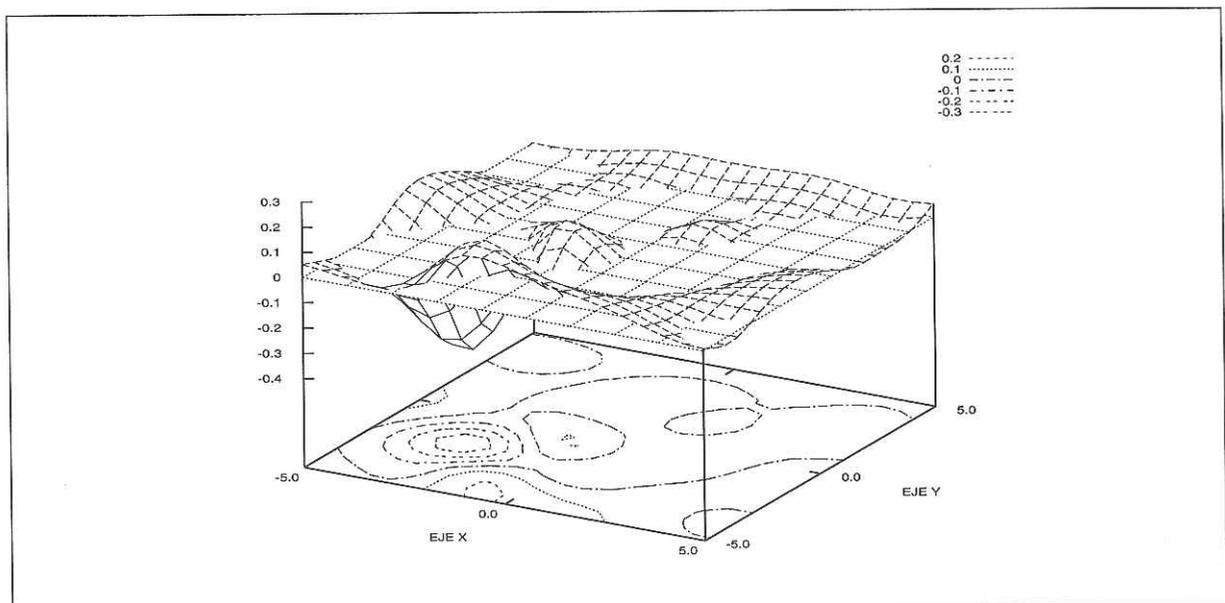


Figura 112: Gráfica del error para $\alpha = 1000.0$; 0.2% de error

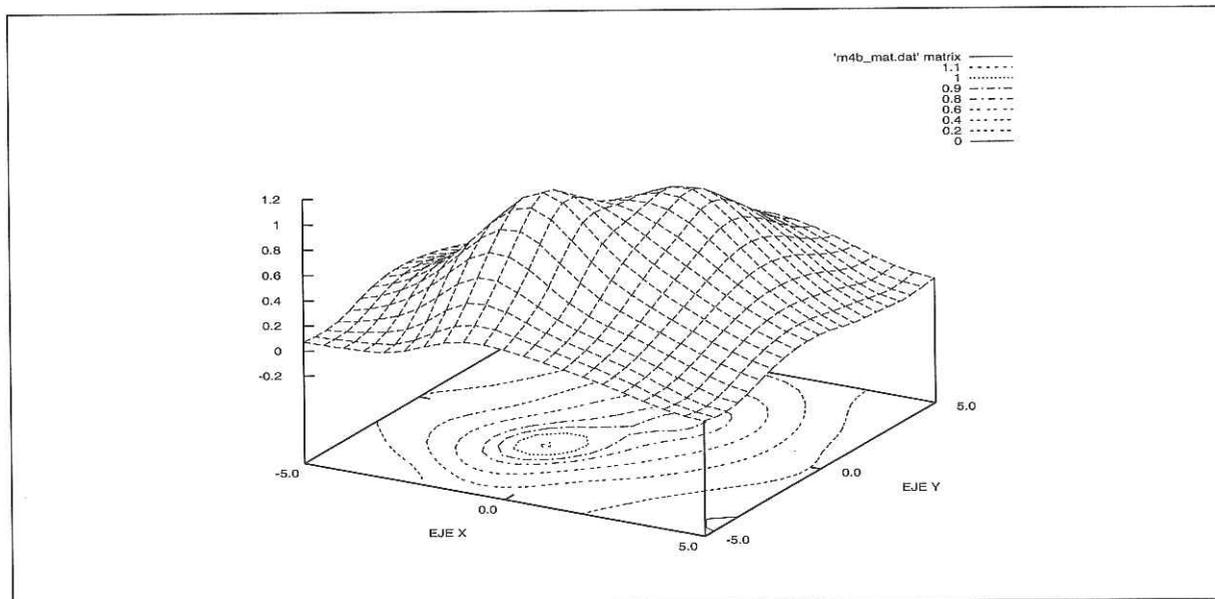


Figura 113: Gráfica de $\hat{m}(r)$ para $\alpha = 10000.0$; 0.2% de error

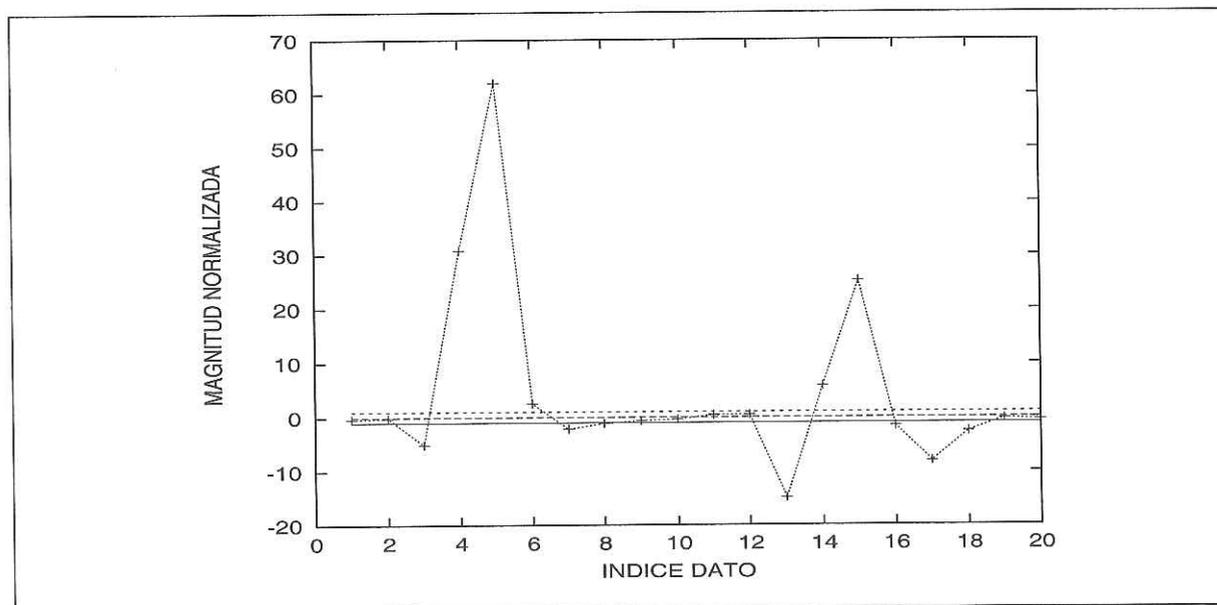


Figura 114: Gráfica de residuales para $\alpha = 10000.0$; 0.2% de error

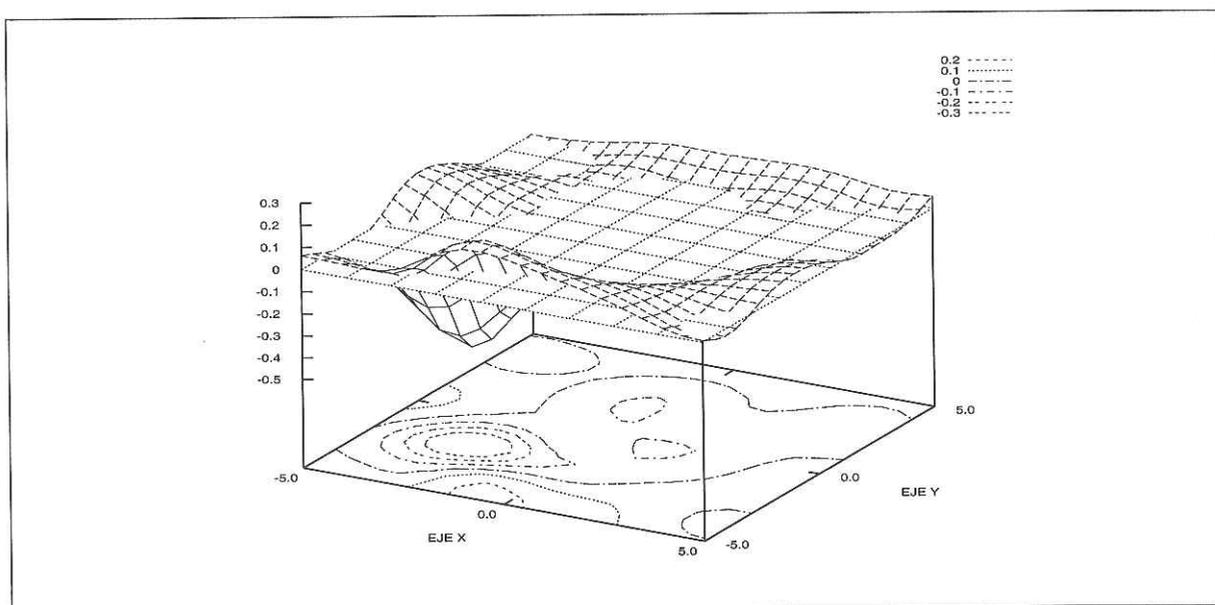


Figura 115: Gráfica del error para $\alpha = 10000.0$; 0.2% de error

III.6 Análisis de Resultados. Conclusiones

Los resultados obtenidos para los distintos niveles de ruido muestran un comportamiento similar al observado en los resultados del problema de validación.

Para valores pequeños del parámetro α ($\approx 10^{-4}$) se observa

- La estimación muestra oscilaciones grandes.
- Los errores tiene magnitudes grandes, del ordel del 30% del valor de la solución verdadera.
- Los residuales son muy pequeños y despreciables.

Para valores grandes de α ($\approx 10^4$) se observa

- La solución tiende al plano constante cero con algunas oscilaciones, es decir, las estimaciones están sobreestabilizadas.
- Los errores tienden a ser del 100%, es decir, corresponden a la solución verdadera con signo contrario.
- Las componentes del vector de residuales son muy grandes y muestran conductas sistemáticas.

En base al análisis de residuales, error de la estimación y estabilidad de la estimación, las estimaciones consideradas óptimas, para cada nivel de ruido, son las siguientes:

a) Para datos con el 0.2% de ruido la solución óptima estimada corresponde a

$$\alpha = 1000.0$$

b) Para datos con el 1.0% de ruido la solución óptima estimada corresponde a

$$\alpha = 1000.0$$

c) Para datos con el 5.0% de ruido la solución óptima estimada corresponde a

$$\alpha = 100.0$$

De nuevo, se observa que la solución estimada contiene oscilaciones que no pertenecen a la solución verdadera. Estas deficiencias parecen deberse al número pequeño de puntos de observación y al hecho de que estos no cubren uniformemente el área de interés. Aún así el procedimiento numérico computacional es adecuado, la función incógnita se recupera y ambos máximos locales se pueden resolver. Suponiendo que no se conoce la solución verdadera (función incógnita), el análisis combinado de residuales y de estabilidad de la solución puede producir una solución óptima que es adecuada.

Capítulo IV

Programas Computacionales

En este capítulo, se describen los programas computacionales utilizados para resolver los problemas inversos definidos en los capítulos II y III. Se acompañan, además, los diagramas de flujo generales para cada programa, una descripción corta de las funciones que se usan en la codificación, se resume el trabajo de optimización y se agrega información acerca de tiempos de ejecución.

IV.1 Características de la Computadora

La plataforma utilizada para desarrollar los programas es una super computadora Origin 2000, identificada con el nombre **cicese2000**, perteneciente al Departamento de Cómputo, Dirección de Telemática del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), que presenta las siguientes características:

- a) Computadora Multiprocesador MIMD de Memoria compartida Origin 2000 de Silicon Graphics.
- b) 10 procesadores MIPS R10000.
- c) Velocidad por procesador de 195 MHz.
- d) Rendimiento máximo teórico por procesador de 390 Mflops (Millones de Operaciones de Punto Flotante por Segundo).
- e) 1.25 GigaByte de memoria RAM.
- f) 40.9 GigaByte de capacidad en disco duro.
- g) Sistema Operativo Silicon Graphics IRIX 6.4 IP27.

Esta máquina es accesada a través de la red Internet, dentro de la red de CICESE, en particular desde una máquina Silicon Graphics O2 ubicada en el Laboratorio de Sismología Computacional, a cargo del director de la tesis.

IV.2 Algoritmos y Codificación

Para resolver cada uno de los problemas inversos de los capítulos dos y tres, se utilizan tres programas

- a) `directo.c`
- b) `matriz.s.c`

c) **invertir.c**

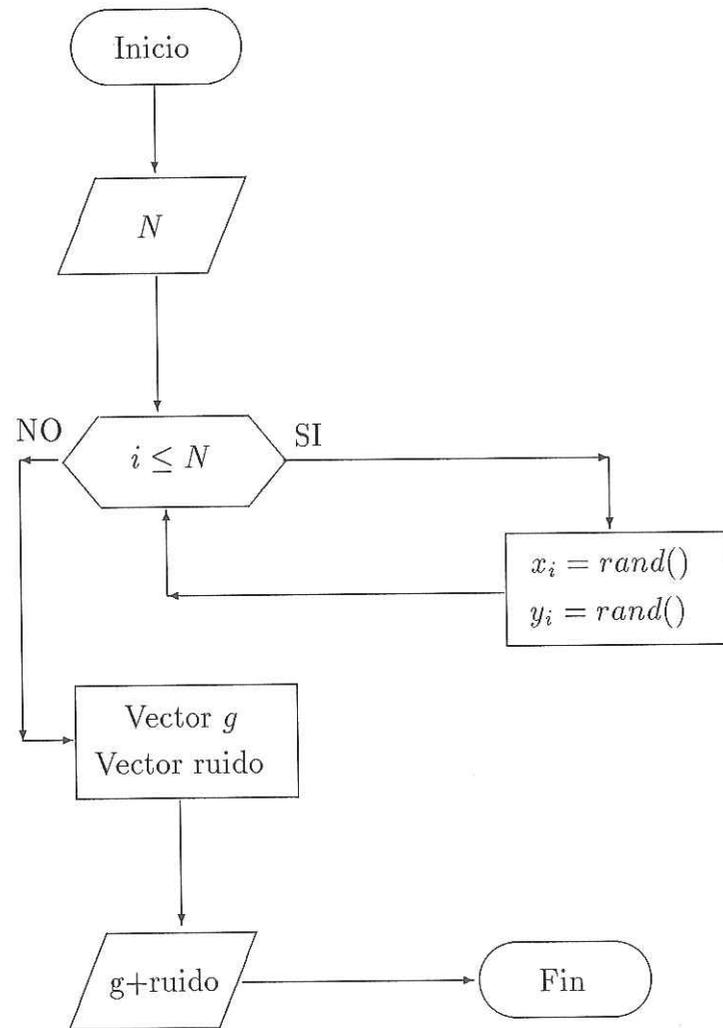
El programa **directo.c** genera los puntos aleatorios donde se calculan los valores de la función observación, produce el vector de datos g , agrega ruido aleatorio a estos valores y genera la matriz de covarianza del error de los datos C_{yy} . El programa **matriz_s.c** calcula los elementos de la matriz S , resolviendo numéricamente una integral cuádruple. El programa **invertir.c** obtiene la estimación $\hat{m}(r)$ para diferentes valores del parámetro de regularización α , mediante la expresión (37). Este programa utiliza los resultados de los programas **directo.c** y **matriz_s.c**.

El código fuente de estos programas fue escrito en lenguaje C siguiendo el estándar ANSI C, definiendo todas las variables de punto flotante en doble precisión. Se evitó la utilización de librerías especiales optimizadas para la plataforma Origin 2000 para que el código sea portable entre diferentes plataformas y compiladores.

A continuación, se describe cada uno de los programas.

IV.2.1 Programa directo.c

El diagrama de flujo del programa **directo.c** es el siguiente:



donde N es el número de datos u observaciones.

Para el problema de validación (Capítulo II) el programa utiliza tres funciones; los valores de la función conocida g se calculan analíticamente.

```
int main(void)
```

```
double gasdev(int *)
```

```
double ran1(int *)
```

En la función *main()* se calculan los puntos aleatorios (x_i, y_i) , se evalúa analíticamente $g(x_i, y_i)$ y se le agrega ruido aleatorio a cada observación utilizando las ecuaciones (56) y (60). Las funciones *gasdev()* y *ran1()* son utilizados para generar una secuencia de números aleatorios con una distribución gaussiana de media cero y desviación estándar uno (Press *et al*, 1992).

El valor del parámetro p que se utiliza en la ecuación (56)

$$r = e \left[(p)(|g_i|) + (0.005)\left(\frac{\gamma}{2.0}\right) \right] \quad (56)$$

y en la expresión (60)

$$r = e \left[(p)(|g_i|) + (0.002)\left(\frac{\gamma}{2.0}\right) \right] \quad (60)$$

se especifican en la función *main()*, y puede tomar los valores de 0.002, 0.01 y 0.05, correspondientes a un error relativo del 0.2%, 1.0% y 5.0%, respectivamente. Para generar datos sin ruido, se especifica $p = 0.0$.

El número de datos que se quieren generar, así como la región del plano donde se van a calcular, se especifican en el archivo de cabecera **invertir.h**

```
#define NUM_DATOS 19  
  
#define XMIN_SOL -5  
  
#define XMAX_SOL 5  
  
#define YMIN_SOL -5  
  
#define YMAX_SOL 5
```

Como salidas, el programa genera los archivos **datos.dat** y **mat_cov.dat**. El archivo de salida **datos.dat** contiene las coordenadas x_i , y_i y el valor $g(x_i, y_i)$ con o sin ruido aleatorio. El archivo **mat_cov.dat** contiene los elementos de la diagonal principal de la matriz de covarianza C_{yy} , ordenados como arreglo.

Para resolver el segundo problema inverso (Capítulo III), el programa utiliza cinco funciones

```
int main(void)  
  
double gasdev(int *)  
  
double ran1(int *)  
  
double evaluar_g(double x_i, double y_i)
```

La evaluación de $g(x_i, y_i)$ se realiza numéricamente mediante la función *evaluar_g()*. Esta función recibe como argumentos las coordenadas del i -ésimo punto (representadas por las variables x_i y y_i); como salida, el valor calculado de $g(x_i, y_i)$. El método de integración que se utiliza es el método de cuatro puntos. Los límites de integración y el paso de integración se definen en el archivo de cabecera **invertir.h**

```
#define XMIN_G -10  
  
#define XMAX_G 10  
  
#define YMIN_G -10  
  
#define YMAX_G 10  
  
#define H_G 0.2
```

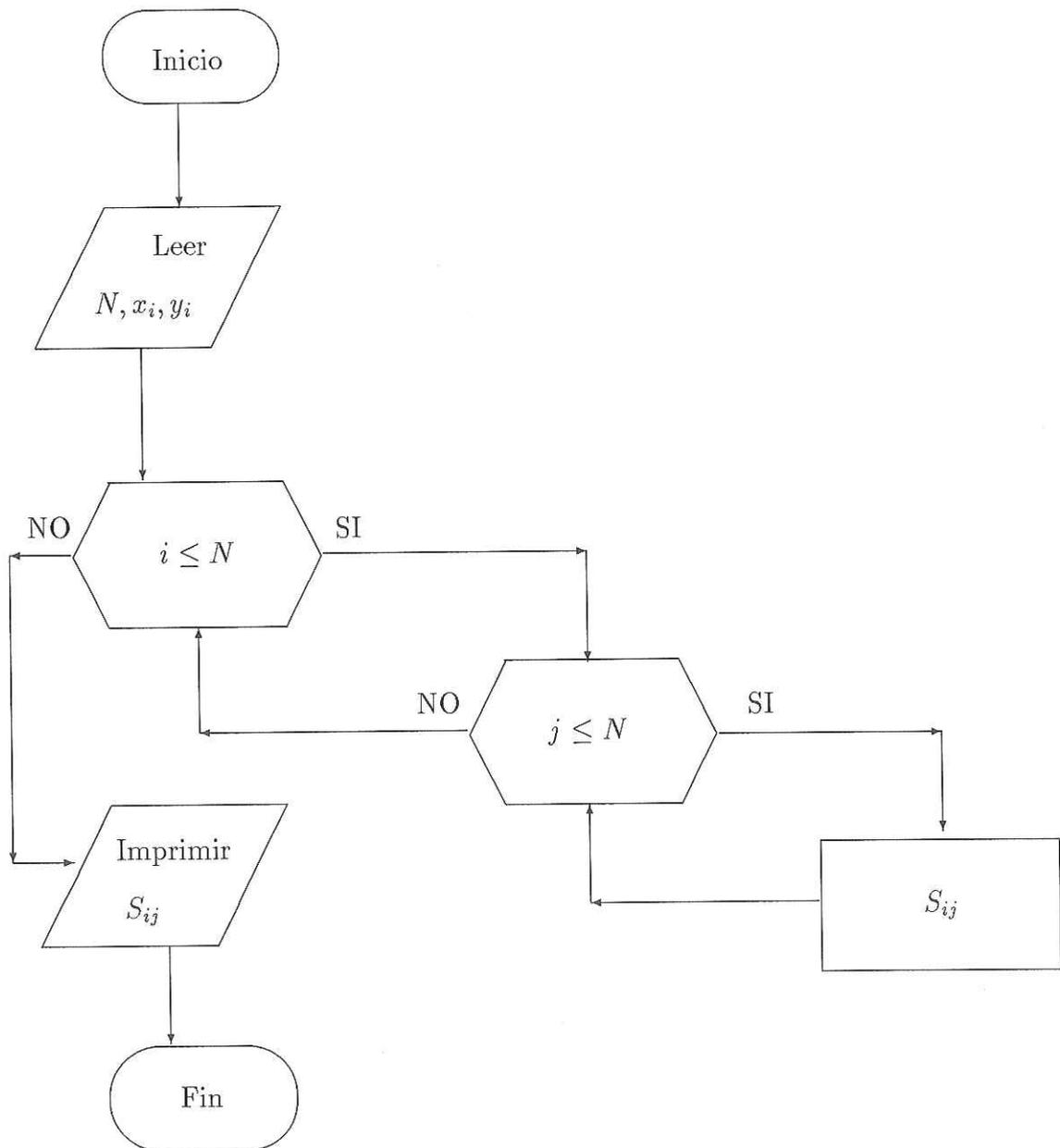
El número de datos que se generan para el segundo problema son veinte, definidos en el archivo `invertir.h`

```
#define NUM_DATOS 20
```

En este programa se utilizan dos ciclos sencillos, uno en que se calculan los puntos aleatorios (x_i, y_i) y otro en el que se calcula el vector g contaminado con ruido aleatorio. El número de veces que se repite cada ciclo es igual al número de observaciones que se quieren generar.

IV.2.2 Programa matriz_s.c

El diagrama de flujo es



donde N es el número de observaciones.

El programa `matriz_s.c` se utiliza en la solución de los dos problemas inversos, ya que el kernel $K_i(r)$ es el mismo para ambos problemas. Sólo cambia el número, valor y coordenadas de las observaciones.

El programa codificado tiene las siguientes funciones

```
int main(void)
```

```
void leer_datos(void)
```

```
double elemento_s(int i,int j)
```

```
double e_i_r(double x0,double y0,double xsing,double ysing)
```

```
double int_2d(double xmin,double xmax,double ymin,double ymax, double x0,double
y0,double xsing,double ysing)
```

La función `leer_datos()` se encarga de leer los puntos (x_i, y_i) del archivo `datos.dat`.

La función `elemento_s()` calcula numéricamente la integral doble

$$S_{ij} = \int K_i(r) e_j(r) dr \quad (55)$$

Tiene como argumentos de entrada los índices i y j , los cuales utiliza para determinar las coordenadas del i -ésimo dato (x_i, y_i) y del j -ésimo dato (x_j, y_j) . La salida de la función es el valor calculado del elemento S_{ij} . La región de integración

$$|x| \leq 9 \quad |y| \leq 9$$

se divide en una malla con celdas cuadradas de lado 0.1 y los nodos de la malla son calculados y sus valores almacenados en la matriz `malla_s[][]` con precisión doble. Para

el cálculo de cada nodo se evalúa el argumento de la integral (55) con las coordenadas correspondientes a cada nodo. Una vez calculados los nodos la malla se divide en celdas cuadradas de lado 0.2 y se utiliza el método de nueve puntos para integrar cada celda.

La función $e_{i-r}()$ calcula numéricamente la expresión

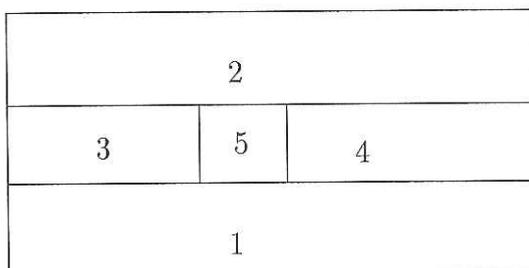
$$e_i(r) = \int K_i(r') G(r, r') dr' \quad (24)$$

donde la r' es la variable de integración. Tiene como argumentos de entrada las coordenadas del i -ésimo punto (x_i, y_i) (indicadas en el código por las variables $x0$ y $y0$) y las coordenadas del punto r (indicadas en el código por las variables $x\text{sing}$ y $y\text{sing}$). La salida de la función es el valor calculado de $e_i(r)$, el cual se utiliza en el cálculo de S_{ij} .

La región de integración es

$$|x| \leq 10 \quad |y| \leq 10$$

y como se indicó en la sección II.6, esta región de integración se divide en cinco subregiones



La subregión cinco incluye el punto singular, siendo cuadrada y de lado 0.2; en esta subregión se calcula analíticamente $e_i(r)$ mediante la ecuación (54). Las subregiones 1,2,3

y 4 se dividen en celdas cuadradas de lado 0.1, y cada una se integra numéricamente utilizando el método de cuatro puntos. La función *int_2d()* lleva a cabo la integración numérica de estas cuatro subregiones.

La función *int_2d()* tiene como argumentos las coordenadas del punto inferior izquierdo (representadas en el código por las variables *xmin* y *ymin*) y del punto superior derecho (representadas en el código por las variables *xmax* y *ymax*) de la subregión que se integra. Cada subregión se divide en celdas cuadradas de lado 0.1 y cada celda se integra con el método de cuatro puntos.

Para el cálculo de la matriz *S* se aprovecha el hecho de que es simétrica ($S_{ij} = S_{ji}$), por lo que se calculan sólo los elementos S_{ij} que cumplen la condición $i \leq j$.

Los resultados del programa **matriz.s.c** son almacenados en el archivo **matriz.s.dat** para su posterior uso. El formato es el siguiente: se imprimen tres columnas de datos, la primera columna indica el valor del índice *i*, la segunda columna el valor del índice *j* y la tercera columna el valor del elemento S_{ij} . La matriz *S* esta ordenada por filas.

Analicemos los ciclos utilizados en este programa. La función *main()* usa dos ciclos anidados en los cuales se calculan los elementos del triangulo superior de la matriz simétrica *S*, por filas. El ciclo interno contiene un llamado a la función *elemento_s()* para calcular un elemento S_{ij} . El índice de cada ciclo corre de 1 a 19, en el problema de validación y de 1 a 20 en el problema de aplicación. Por tanto en el problema de validación se llama 190 veces a la función *elementos_s()*, mientras que en el problema de aplicación se llama en 210 ocasiones.

En la función *elemento_s()* se utilizan dos ciclos anidados para el cálculo de la matriz *malla_s[][]*. El índice de ambos ciclos corre de 1 a 181, por lo que el ciclo interno es ejecutado 32761 veces. El ciclo interno contiene un llamado a la función *e_i_r()* para el cálculo de cada nodo. Cuando se calculan los nodos de la malla, se utilizan dos ciclos anidados para calcular S_{ij} . Cada ciclo corre de 1 a 90, por lo que el ciclo interno se calcula 8100 veces.

La función *e_i_r()* resuelve la doble integral

$$e_i(r) = \int K_i(r')G(r, r')dr' \quad (24)$$

dividiendo la región de integración

$$|x| \leq 10 \quad |y| \leq 10$$

en cinco subregiones. Una de las subregiones, cuadrada de lado 0.2, se resuelve analíticamente y las subregiones restantes se resuelven numéricamente mediante la función *int2d()*, en la cual se utilizan dos ciclos anidados para la integración numérica de cada subregión, usando el método de cuatro puntos. De esta forma, podemos considerar que la función *e_i_r()* usa dos ciclos anidados donde el índice de cada ciclo corre de 1 a 200, por lo que el número de veces que se llama al ciclo interno es de 40,000.

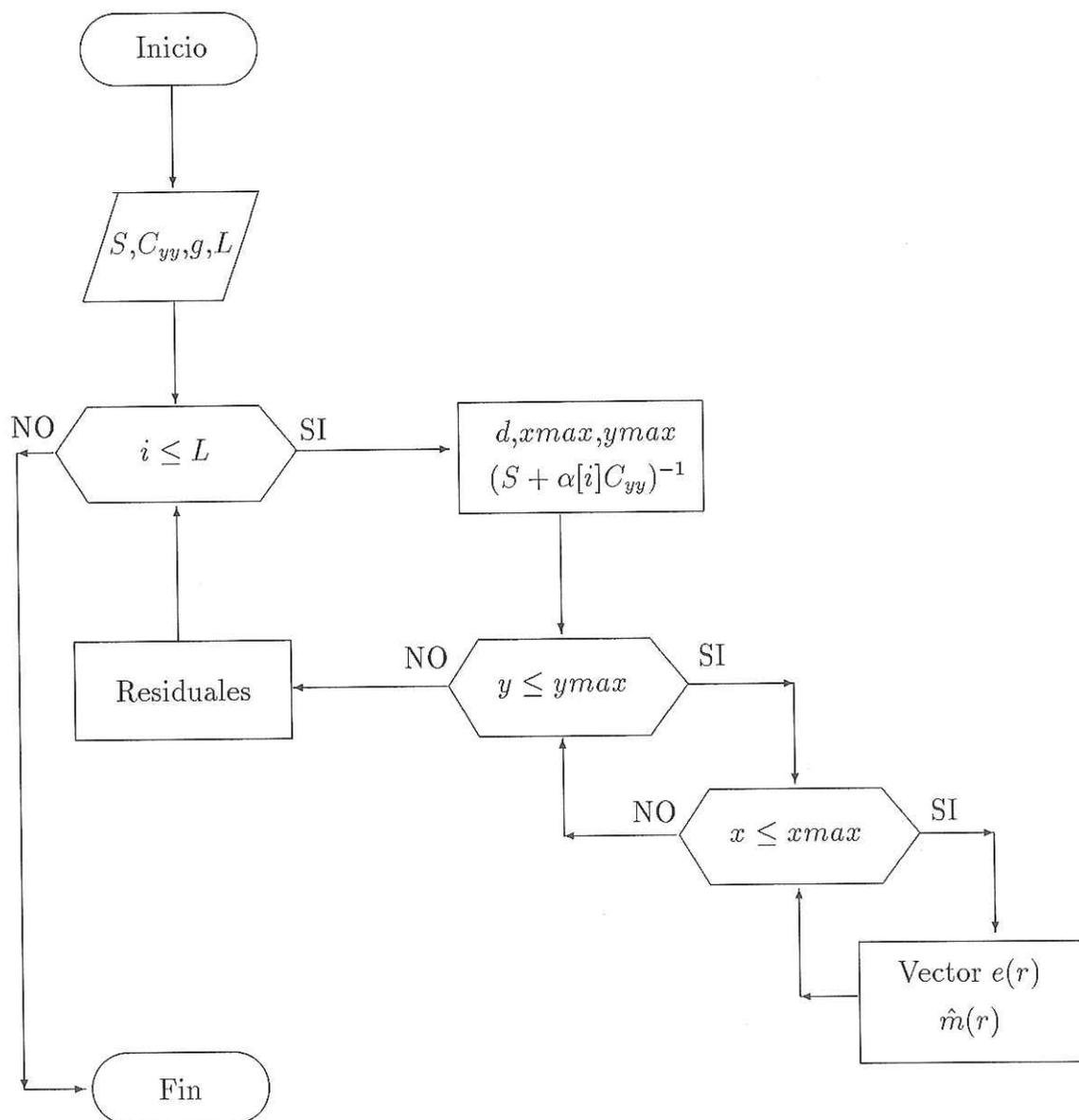
Por lo tanto el número de veces que se calcula el ciclo más interno, considerando los cuatro ciclos anidados, para calcular un elemento de la matriz S es

$$(40000) (32761) = 1.31044 \times 10^9$$

Como se observa el costo en tiempo de cálculo de la matriz S es muy alto, ya que se debe considerar que en cada ciclo se realizan operaciones aritméticas, como sumas, multiplicaciones y uso de las funciones $exp()$ y $log()$.

IV.2.3 Programa invertir.c

El diagrama de flujo para este programa es el siguiente



donde x_{min} , x_{max} , y_{min} y y_{max} delimitan la región donde se calcula la estimación

$\hat{m}(r)$ y L es el número de valores de α .

El programa **invertir.c** obtiene la estimación $\hat{m}(r)$ para diferentes valores del parámetro de regularización α , mediante la expresión (37). El programa tiene tres archivos de entrada:

datos.dat Este archivo contiene los puntos (x_i, y_i) y el vector de datos g .

mat_cov.dat Este archivo almacena los elementos de la diagonal principal de la matriz de covarianza C_{yy} .

matriz_s.dat Este archivo guarda los elementos de la matriz S .

Una vez leídos estos archivos, se aplica la expresión (37) para calcular la estimación para diferentes valores de α . Los valores de α están definidos dentro del programa principal, como un vector.

El programa codificado tiene ocho funciones:

```
int main(void)
```

```
void leer_datos(void)
```

```
void leer_mat_cov(void)
```

```
void leer_mat_s(void)
```

```
void calcular_inversa(void)
```

```
double e.i.r(double x0,double y0,double xsing,double ysing)
```

```
double int_2d(double xmin,double xmax,double ymin,double ymax, double x0,double y0,double xsing,double ysing)
```

```
void gaussj(double **a,int n,double **b,int m)
```

Las funciones *leer_datos()*, *leer_mat_cov()* y *leer_mat_s()* se utilizan para leer los archivos **datos.dat**, **mat_cov.dat** y **matriz_s.dat**, respectivamente.

La función *calcular_inversa()* calcula la matriz

$$(S + \alpha C_{yy})^{-1}$$

utilizando para ello la función *gaussj()* la cual calcula la matriz inversa de la matriz correspondiente (Press *et al*, 1992). Aunque existen métodos más eficientes para calcular la matriz inversa de una matriz simétrica como son los métodos de tipo Choleski, ellos no son utilizados ya que las dimensiones de la matriz $(S + \alpha C_{yy})$ no son de gran tamaño, siendo de 19x19 en el problema de validación y 20x20 en el problema de aplicación. La misma consideración justifica que, en vez de calcular la matriz inversa, se resuelva en forma directa el sistema de ecuaciones.

La función *e_i_r()* calcula los elementos del vector $e(r)$ para cada punto r donde se desea calcular la estimación, utilizando la función *ind_2d()*. Las características de estas dos funciones son las mismas que se dieron en la sección anterior, ya que se utilizan en el programa **matriz_s.c**.

La estimación $\hat{m}(r)$ se calcula en los nodos de una malla formada por celdas cuadradas de lado $h = 0.5$, en la región

$$|x| \leq 8 \quad |y| \leq 8$$

por lo que se calculan en total 32x32 estimaciones $\hat{m}(r)$. Esta estimación se almacena en una matriz para utilizarse, después de mostrarla en pantalla, en el cálculo del vector

de residuales.

El i -ésimo elemento del vector de residuales, correspondiente al i -ésimo dato, se obtiene utilizando la expresión

$$r_i = g_i - \int K_i(r) \hat{m}(r) dr$$

La integración se hace numéricamente, ya que $\hat{m}(r)$ sólo es conocida en una malla discreta de puntos, utilizando el método de nueve puntos. La integración se hace sobre la región donde se calcula la estimación $\hat{m}(r)$.

Para cada valor de α el programa produce los siguientes resultados

- a) El valor de la estimación para diferentes puntos junto con el error en la estimación, en una tabla cuyas columnas son

$$x \quad y \quad m(x, y) \quad \hat{m}(x, y) \quad \hat{m}(x, y) - m(x, y)$$

- b) El vector de residuales, indicando el índice correspondiente a cada residual.

En este programa se realizan algunas operaciones elementales (multiplicación y suma) entre matrices y vectores, calculos que se tienen que realizar para la evaluación de la expresión (37). El archivo `matvec.c` contiene una serie de funciones que permiten realizar estas operaciones entre matrices y vectores.

En la función `main()` el ciclo más externo determina el valor del parámetro α con el cual se hace la estimación. El índice de este ciclo corre de 1 a 8 en el problema de validación y de 1 a 12 en el problema de aplicación. Dentro de este ciclo se encuentran

dos ciclos anidados, con los cuales se calcula la estimación $\hat{m}(r)$ en los nodos de la malla definida con anterioridad. Cada ciclo corre de 1 a 33. Por lo tanto el número de puntos donde se hace la estimación es de 33×33 y es el número de veces que se calcula el ciclo interno para un solo valor de α .

Dentro del ciclo para α , se utilizan tres ciclos anidados para el cálculo de los residuales. El ciclo más externo controla el índice del residual que se va a calcular y su índice corre de 1 a 19 en el problema de validación y de 1 a 20 en el problema de aplicación. En los dos ciclos anidados internos se hace el cálculo de los elementos del vector de residuales

$$r_i = g_i - \int K_i(r) \hat{m}(r) dr$$

usando el método de nueve puntos para el cálculo de la doble integral. El índice de cada ciclo corre de 1 a 16. Por lo tanto para un solo valor de α el ciclo interno se calcula $19 \times 16 \times 16$ en el problema de validación y $20 \times 16 \times 16$ en el problema de aplicación

IV.3 Optimización

Los programas **directo.c**, **matriz.s.c** e **invertir.c** se optimizaron en tres etapas sucesivas:

- a) Se examina cada uno de los ciclos; las operaciones que no dependen del índice correspondiente son puestas fuera del ciclo.
- b) Se ordenan y simplifican expresiones, para reducir el número de operaciones "caras", comúnmente multiplicaciones y divisiones.
- c) Se reducen las llamadas a funciones, principalmente dentro de los ciclos más internos.

No se llevó a cabo una optimización de los recursos de memoria, ya que los programas no utilizan una gran cantidad de ella. El programa **matriz.s.c** utiliza la mayor cantidad de memoria de los tres programas, usando alrededor de 270Kb de memoria, utilizada casi en su totalidad por la matriz *malla_s[][][]*. El programa **invertir.c** utiliza aproximadamente 50 Kilobyte, siendo menor la utilizada por el programa **directo.c**.

Una vez que se aplica la optimización al código, los programas se compilaron sin utilizar las opciones de compilador para optimización automática, utilizando las siguientes opciones

```
cc [archivo(s)] -lm -n32 -mips4 -r10000
```

donde

-lm Opción que permite utilizar las librerías matemáticas.

-n32 -mips4 Estas opciones aseguran que se genere código máquina que tenga acceso a un espacio de direcciones de 32bit, así como para utilizar los registros de memoria y los protocolos de llamada a subrutinas que son mejores que los implementados en versiones anteriores del procesador.

-r10000 Esta opción indica al compilador el tipo de procesador en el que debe ejecutarse el código, y así hacer la selección adecuada de librerías matemáticas durante la compilación.

Un punto importante de aclarar es respecto a las opciones por default del compilador. Las opciones por default se indican en el archivo */etc/compiler.defaults*. Como este archivo esta bajo control del administrador del sistema no se puede estar seguro de las opciones por default del compilador, por lo que es preferible indicar las opciones más importantes personalmente. En particular en la computadora **cicese2000** las opciones por default son **-n32 -mips4 -r10000**, pero para una mayor seguridad se indicaron en las opciones del compilador. Con respecto al nivel de optimización, por default es **-O0** con la cual se desactiva toda optimización. Una mayor información de las opciones del compilador se encuentra en el manual del compilador cc (accesible a través de la consola).

El tiempo de ejecución del programa **directo.c** para el cálculo de 19 datos en el problema de validación y de 20 datos en el problema de aplicación, es menos que 3 segundos. El tiempo de ejecución para el programa **matriz.s.c** en el cálculo de un elemento de la matriz S es de aproximadamente 37 minutos. El tiempo de ejecución del

programa **invertir.c** en el cálculo de la estimación $\hat{m}(r)$ y del vector de residuales para un valor del parámetro α es de aproximadamente 26 minutos. Como los tiempos de ejecución de los programas **matriz.s.c** e **invertir.c** son grandes, se utilizó enseguida la optimización automática utilizando las siguientes opciones del compilador

```
cc [archivo(s)] -lm -n32 -mips4 -r10000 -LNO -IPA=on -O3
```

donde

- LNO** Esta opción indica al compilador que realice una optimización automática sobre los ciclos, mediante reordenamiento o fusión de los mismos. Además busca optimizar el uso de la memoria caché (memoria mucho más rápida que la memoria RAM y utilizada directamente por el procesador para realizar las operaciones).
- IPA=on** Esta opción activa el Analizador Inter-Procedural, el cual analiza la manera en que los procedimientos interactúan dentro del programa para modificar esta interacción en busca de un mejor rendimiento.
- O3** Esta opción permite que se lleve a cabo el nivel más alto de optimización por parte del compilador.

Con esta optimización, el tiempo de ejecución del programa **matriz.s.c** se redujo a 16 minutos por elemento de la matriz S , mientras que el tiempo de ejecución del programa **invertir.c** se redujo a 10 minutos para cada valor de α , es decir se obtuvo una reducción del 56% y del 60%, respectivamente, en el tiempo de ejecución.

Es importante señalar que los tiempos de ejecución señalados corresponden al uso de un solo procesador de la computadora **cicese2000**.

Para reducir aún más el tiempo de ejecución del programa **matriz.s.c**, se hace una paralelización del código. Existen dos opciones para realizar esta paralelización. Una es la paralelización automática por medio del compilador, utilizando opciones de compilación, y la otra es incluir directivas de compilación en el código junto con la opción **-mp** al momento de compilar. Las directivas de compilación utilizadas forman parte del lenguaje Power C, que es una extensión del lenguaje C que permite paralelizar en los sistemas **Origin 2000**. Las directivas son ignoradas si la opción **-mp** no es indicada en la compilación.

Inicialmente se intentó una paralelización automática por medio de opciones del compilador, pero no se obtuvieron resultados. Mediante las directivas de Power C se puede llevar a cabo la paralelización de ciclos sin dependencia de datos de manera relativamente sencilla y es esta clase de ciclos los que se paralelizan en el programa **matriz.s.c**. La paralelización se aplica en la función *elemento_s()*, sobre el ciclo externo utilizado en el cálculo de la matriz *malla_s[][]*. De esta manera la matriz *malla_s[][]* se calcula por bloques, cada uno de los cuales en forma independiente y en diferentes procesadores.

A continuación se muestra el código de los ciclos anidados que se paralelizaron, dentro de la función *elemento_s()*, junto con las directivas utilizadas.

```
#pragma parallel

#pragma pfor iterate(p=1;intervalos_y+1;1)

#pragma local(p,q,cy1,x,y) byvalue(intervalos_x,intervalos_y)

#pragma shared(malla_s,datos,intervalos_x,intervalos_y,h_int)

{

for(p=1;p<=intervalos_y+1;++p){

    y=-9.0+(h_int)*(p-1);

    cy1=exp(-((y-datos[i].y0)*(y-datos[i].y0)));

    for(q=1,x=-9.0;q<=intervalos_x+1;++q,x+=h_int){

        malla_s[p][q]=cy1*

        exp(-((x-datos[i].x0)*(x-datos[i].x0)))*

        e_i_r(datos[j].x0,datos[j].y0,x,y);

    }

}

}
```

La directiva *parallel* indica que se inicia una región de paralelización. La directiva *pfor* indica al compilador el número de veces que se ejecuta el ciclo a paralelizar, y cuyo número de ejecuciones se va a repartir entre los distintos procesadores. La directiva *local* indica cuales son variables locales en cada proceso y que no son utilizadas por los otros procesadores. Y la directiva *shared* indica las variables que van a ser accedidas por los diferentes procesadores.

Los tiempos que se obtienen en el cálculo de un elemento de la matriz S , utilizando varios procesadores, se muestran en la tabla XI.

Tabla XI: Tiempos de ejecución en paralelo de **matriz_s.c**

<i>Num. Procesadores</i>	<i>Tiempo (Min:Seg)</i>
2	7:56
3	5:22
4	4:05
5	3:15
6	2:43
7	2:19
8	2:01

Como se observa, los resultados muestran que el tiempo disminuye en forma inversamente proporcional al número de procesadores, siendo la constante el valor de 16 minutos

$$t_n = 16.0/n$$

donde n es el número de procesadores. Este comportamiento se debe a que el ciclo que se paraleliza no tiene dependencia de datos.

IV.4 Conclusiones

En este capítulo se describen los programas **directo.c**, **matriz.s.c** e **invertir.c**, los cuales se utilizan en la solución de los problemas inversos de los capítulos II y III. Se hace una descripción de cada una de las funciones que se utilizan en cada programa. Para resolver un problema inverso primero se ejecuta el programa **directo.c**, después el programa **matriz.s.c** y por último el programa **invertir.c**.

Los programas son optimizados en tres etapas: primero se les aplica una optimización manual, en la cual se analiza el código y se modifica si es necesario para reducir, principalmente, los tiempos de cómputo; después, se hace una optimización automática por medio de opciones del compilador; por último, se paraleliza el programa **matriz.s.c**. Para cada etapa, se muestran ejemplos de tiempos de cálculo.

La mayor parte del tiempo de cómputo para resolver un problema inverso se utiliza en el cálculo de la matriz S . En el caso de utilizar un solo procesador y aplicando la optimización mediante el compilador, el tiempo total de cómputo para resolver el problema de validación es de aproximadamente 3120 minutos, de los cuales 3040 minutos se utilizan para el cálculo de 190 elementos de la matriz S ; en el problema de aplicación, el tiempo total de cómputo es de 3480 minutos, de los cuales 3360 minutos se utilizan en el cálculo de 210 elementos de la matriz S .

Mediante la paralelización se logran reducir los tiempos de cómputo de la matriz S . Utilizando 8 procesadores, se logra calcular un elemento de la matriz S en 2 minutos. Por lo tanto, en el problema de validación, se calculan los 190 elementos de la matriz

S en 380 minutos, mientras que los 210 elementos de S para el problema de aplicación se calculan en 420 minutos. Para el problema de validación el tiempo total de cómputo es de 460 minutos, mientras que para el problema de aplicación el tiempo es de 540 minutos.

Debido a que los tiempos de ejecución para el cálculo de la matriz S son bastante considerables, existe una fuerte restricción en cuanto al número de observaciones que se pueden utilizar, ya que si se tienen N observaciones, el número de elementos a calcular es de $N(N + 1)/2$, ya que se calcula sólo el triángulo superior de la matriz simétrica S .

Una alternativa para buscar reducir los tiempos de cómputo es explorar diferentes métodos para la integración numérica de los elementos S_{ij} , como son los métodos de Monte Carlo. Otra alternativa es reducir los pasos de integración con lo cual disminuye el tiempo de cómputo pero también la precisión de los resultados.

Por último, los programas desarrollados para el trabajo de esta tesis pueden ser modificados fácilmente para resolver otros problemas inversos. En el caso de contar con datos de campo no se necesita el programa **directo.c**; para el caso de que el problema tenga como kernel una función distinta, la función que se habrá que modificar es $e_{i-r}()$ tanto en el programa **matriz.s.c** como en **invertir.c**. Si se modifican los métodos de integración, las funciones a modificar son $e_{i-r}()$ y $int_{2d}()$.

Capítulo V

Conclusiones y Perspectivas

V.1 Conclusiones

Este trabajo de tesis consistió en el desarrollo de un procedimiento matemático para resolver el problema inverso de una ecuación de Fredholm de primera clase bivalente semidiscreta

$$g_i = \int K_i(r) m(r) dr \quad (19)$$

utilizando para ello un método de regularización. Para ello se generaliza el procedimiento desarrollado por Yanovskaya y Ditmar (Yanovskaya y Ditmar, 1990), minimizando la norma cuadrática de los residuales y la norma cuadrática del gradiente de $m(r)$

$$\min \left\{ \|g - Km\|_{C_{yy}}^2 + \alpha \int |\nabla m(r)|^2 dr \right\} \quad (20)$$

y desarrollar la solución en términos de la función de Green $G(r, r')$ del laplaciano en

2-D. El estimador obtenido es

$$\hat{m}(r) = e^T(r)(S + \alpha C_{yy})^{-1}g + \frac{1 - e^T(r)(S + \alpha C_{yy})^{-1}d}{d^T(S + \alpha C_{yy})^{-1}d}d^T(S + \alpha C_{yy})^{-1}g \quad (37)$$

donde d y $e(r)$ son vectores y S es una matriz. Para el cálculo de los elementos de los vectores d y $e(r)$ es necesario resolver integrales dobles, mientras que para el cálculo de cada elemento de la matriz S es necesario resolver una integral cuádruple.

Obtenido el estimador, se desarrollaron los algoritmos y códigos necesarios para la implementación computacional del mismo. Se desarrollaron tres programas escritos en lenguaje C: **directo.c**, el cual genera el vector de observaciones g ; **matriz.s.c**, el cual genera la matriz S ; y, por último, el programa **invertir.c**, el cual calcula la estimación $\hat{m}(r)$ en la región de interés, así como los vectores de residuales y la matriz de error de la solución.

Una vez desarrollados los programas computacionales, se realizó una optimización de cada programa, para lo cual se identificaron y analizaron los ciclos anidados que se utilizan en cada programa. Con las optimizaciones de los ciclos, se obtuvieron los siguientes tiempos. El tiempo de ejecución del programa **directo.c** para el cálculo de 19 y 20 observaciones es menos que 3 segundos. El tiempo de ejecución para el programa **matriz.s.c** en el cálculo de un elemento de la matriz S es de aproximadamente 37 minutos. El tiempo de ejecución del programa **invertir.c**, en el cálculo de la estimación para un valor de α , es de aproximadamente 26 minutos.

Dado que los tiempos de ejecución de los programas **matriz.s.c** e **invertir.c** son altos, se hizo una optimización automática por medio del compilador, con lo cual se

mejoró el rendimiento de los programas. Con esta optimización el tiempo de ejecución del programa **matriz_s.c** se redujo a 16 minutos por elemento de la matriz S , mientras que el tiempo de ejecución del programa **invertir.c** se redujo a 10 minutos en el cálculo de la estimación para un valor de α , es decir se obtuvo una reducción del 55% y del 60%, respectivamente, en el tiempo de ejecución.

Para reducir aún más los tiempos de ejecución del programa **matriz_s.c** se realizó una paralelización del mismo. Para esto, se agregaron directivas de compilación en el código. Estas directivas para paralelizar forman parte del lenguaje Power C, que es una extensión del lenguaje C que permite la paralelización de código en los sistemas **Origin 2000**. Específicamente la paralelización se llevo a cabo sobre uno de los ciclos de la función *elemento_s()*, ya que este presenta independencia de datos. Los tiempos de ejecución obtenidos para el cálculo de un elemento de la matriz S , y para diferente número de procesadores, se muestran a continuación

<i>Num. Procesadores</i>	<i>Tiempo (Min:Seg)</i>
2	7:56
3	5:22
4	4:05
5	3:15
6	2:43
7	2:19
8	2:01

A pesar de lograr reducir el tiempo de cómputo para el cálculo de un elemento de la matriz S , el tiempo de cómputo para el cálculo de la matriz S es alto, ya que si se tienen N observaciones el número de elementos de la matriz S que se deben calcular

es de $N(N + 1)/2$, por lo cual existen fuertes restricciones en el uso del procedimiento computacional cuando se utilizan un gran número de observaciones.

Con los programas computacionales se resolvieron dos problemas inversos con solución matemática dada: un problema de validación, donde el problema directo tiene solución analítica, y un problema de aplicación donde el problema directo se resuelve numéricamente.

En ambos problemas, se realiza la inversión usando observaciones sin ruido y observaciones con ruido gaussiano. Los niveles de ruido relativo utilizados son 0.2%, 1.0% y 5.0%. Para cada nivel de ruido relativo, se hace la estimación $\hat{m}(r)$ para diferentes valores del parámetro de regularización α , siendo ocho valores en el problema de validación y doce valores en el problema de aplicación.

Para cada estimación se calculó el vector de residuales y el error de la solución estimada; mediante el análisis de ambas se determinó la solución óptima correspondiente a los diferentes niveles de ruido. El análisis consiste en determinar un nivel de oscilación (aleatorio) de los residuales que sea consistente con el modelo de errores estimados que tengamos, así como un comportamiento estable (suave) de la solución lo que implica errores de la solución estimada pequeños y aleatorios.

Para ambos problemas inversos se logró obtener una solución óptima que reproduce bien las características de la solución verdadera, con errores pequeños y oscilantes, mientras que el nivel de oscilación de los residuales corresponde estadísticamente al del ruido incluido en el cálculo directo. La presencia de unos pocos errores y residuales que sobrepasan los niveles esperados de deben seguramente al bajo muestreo observacional,

en número y posición, en algunas partes del área que se analiza.

V.2 Perspectivas

Como posibles líneas de trabajo a futuro, para ampliar y mejorar este procedimiento computacional se sugiere:

- a) Para reducir aún más los tiempos de ejecución en el cálculo de la matriz S , se pueden explorar métodos alternativos para la integración numérica en 2-D, como son los métodos de tipo Monte Carlo (Landau y Páez, 1997), y aplicar extensivamente la paralelización por introducción de directivas en el código.
- b) Comparar este procedimiento con otros ya existentes y analizar sus ventajas y desventajas con respecto a éstos para funciones de prueba conocidas.
- c) Aplicar el procedimiento computacional a datos empíricos, esto es, a problemas inversos donde la solución es desconocida, y comparar resultados con los obtenidos con programas (comerciales) en uso en la División de Ciencias de la Tierra, en CICESE.

Literatura Citada

- Abramowitz, M. e I.A. Stegun.** 1970. *Handbook of mathematical functions with formulas, graphs and mathematical tables.* National Bureau of Standards. Novena impresión. 1046p.
- Andrews, H.C. y B.R. Hunt.** 1977. *Digital Image Restoration.* Prentice Hall. New Jersey. 238p.
- Arfken, George.** 1985. *Mathematical Methods for Physicists.* Third Edition. Academic Press Inc. 985p.
- Craig, J.D.Ian y John C. Brown.** 1986. *Inverse Problems in Astronomy. A Guide to Inversion Strategies for Remotely Sensed Data.* Adam Hilger Ltd. Bristol. 152p.
- Deif, A.S.** 1991. *Advanced Matrix Theory for Scientists and Engineers.* Abacus Press. 302p.
- Dimri, Vijay.** 1992. *Deconvolution and Inverse Theory. Application to Geophysical Problems.* Methods in Geochemistry and Geophysics. 29. Elsevier Science

Publishers. 230p.

- Dongarra, J.J. and David W. Walker.** 1995. *Software Libraries for Linear Algebra Computations on High Performance Computers*. SIAM Review. 37(2):151-180p.
- Engl, H.W., M. Hanke y A. Neubauer.** 1996. *Regularization of Inverse Problems*. Kluwer Academic Publishers. 319 p.
- Esparza, F.J. y Frez, J.** 1992. *Regionalización de Velocidades de Ondas Superficiales mediante Condiciones de Continuidad*. Geofísica Internacional. 31(3):263-277p.
- Frez, J.** 1999. *Un Método de Regularización para Invertir la Ecuación Integral de Fredholm de Primera Clase*. Artículo enviado a publicación.
- Greenberg, M.D.** 1971. *Application of Green's functions in science and engineering*. Prentice Hall. 141p.
- Hager, William W.** 1988. *Applied Numerical Linear Algebra*. Prentice Hall. New Jersey. 424p.
- Hansen, Per Christian.** 1992. *Numerical Tools for Analysis and Solution of Fredholm Integral Equations of the First Kind*. Inverse Problems, 8(6):849-872p.
- Hansen, Per Christian.** 1992. *Analysis of Discrete Ill-Posed Problems by Means of the L-Curve*. SIAM Review. 34(4):561-580p.

- Landau, R.H. y M.J.Páez. 1997. *Computational Physics. Problem solving with computers*. John Wiley and Sons. New York. 520p.
- Lanczos, C. 1961. *Linear Differential Operators*. Van Nostrand. London.
- Lawson, C.L. y R.J. Hanson. 1974. *Solving least squares problems*. Prentice Hall. 340p.
- Leithold, Louis. 1992. *El Cálculo con Geometría Analítica*. Sexta Edición. Harla México. 1563p.
- Maron, Melvin J., y Robert J. López. 1995. *Análisis Numérico. Un Enfoque Práctico*. Tercera Edición. CECSA México. 780p.
- Press, W.H., S.A. Teukolsky, W.T. Vetterling y B.P. Flannery. 1992. *Numerical Recipes in C*. Cambridge University Press. Segunda edición. 967p.
- Rowbotham, Peter S. and Pratt, R. Gerhard. 1997. *Improved Inversion Through Use of the Null Space*. Geophysics. 62(3):869-883p.
- Stewart, G.W. 1973. *Computer science and applied mathematics. Introduction to matrix computations*. Academic Press. 441p.
- Swanger, H.J., W.L. Rodi y J.B. Minster. 1983. *An interactive system for two-dimensional magnetotelluric interpretation*. Fifty-third Annual Meeting of Society of Exploration Geophysicist. Las Vegas, Nevada.

- Tikhonov, A.N. y V.Y. Arsenin. 1977. *Solutions of ill-posed problems*. Wiley. Washington.
- Tikhonov, A.N. y A.V. Goncharky. 1987. *Ill-Posed Problems in the Natural Sciences*. Translated by M. Bloch. MIR Publishers. Moscow. 344p.
- Traslosheros, J.C.V., J. Frez, A. Madrid y C. Rebollar. 1990. *Aplicación de la Aproximación Circular en la Estimación de Estructuras Sísmicas Bidimensionales*. Geofísica Internacional. 29(4):211-236p.
- Twomey, S. 1977. *Introduction to the mathematics of inversion in remote sensing and indirect measurements*. Elsevier Scientific Publishing Company. Primera Edición. Amsterdam. 243p.
- Watson, G.A. 1980. *Aproximation Theory and Numerical Methods*. John Wiles and Sons. 229p.
- Weinstock, Robert. 1974. *Calculus of Variations, with Applications to Physics and Engineering*. Dover Publications Inc. New York. 326p.
- Yanovskaya, T.B. y P.G. Ditmar 1990. *Smoothness Criteria in Surface Wave Tomography*. Geophys. J. Int. 102:63-72p

Apéndice A

Del Cálculo de Variaciones sabemos que, en general, si queremos determinar una función $f(x, y)$ que minimize una funcional

$$U(f) = \int \int F(x, y, f, f_x, f_y) dx dy$$

sujeta a la contricción de igualdad

$$\int \int G(x, y, f, f_x, f_y) dx dy = 0$$

siendo $f_x = \partial f / \partial x$ y $f_y = \partial f / \partial y$, debemos resolver la ecuación de Euler-Lagrange

$$\frac{\partial J}{\partial f} - \frac{\partial}{\partial x} \frac{\partial J}{\partial f_x} - \frac{\partial}{\partial y} \frac{\partial J}{\partial f_y} = 0$$

donde

$$J = F + \lambda G$$

siendo λ un multiplicador de Lagrange.

En nuestro problema, se busca una función $m(x, y)$ que minimize la funcional

$$F = \int \int \left[g(s, t) - \int \int K(x, y, s, t) m(x, y) dx dy \right]^2 ds dt +$$

$$\alpha \iint |\nabla m(x, y)|^2 dx dy \quad (61)$$

siendo $|\nabla m(x, y)|$ la magnitud del gradiente de la función $m(x, y)$. Por el Método Variacional, $m(x, y)$ debe satisfacer la ecuación de Euler-Lagrange

$$\frac{\partial F}{\partial m(x, y)} - \frac{\partial}{\partial x} \frac{\partial F}{\partial m_x} - \frac{\partial}{\partial y} \frac{\partial F}{\partial m_y} = 0 \quad (62)$$

Se procede a resolver el primer término de la ecuación (61), por lo que se derivan los términos de la funcional que involucren a la función $m(x, y)$. Siguiendo la convención que $r = (x, y)$, $r' = (s, t)$, $dr = dx dy$ y $dr' = ds dt$, tenemos

$$\frac{\partial F}{\partial m(r)} = \frac{\partial}{\partial m(r)} \int \left[g(r') - \int K(r, r') m(r) dr \right]^2 dr'$$

desarrollando el binomio al cuadrado,

$$\frac{\partial F}{\partial m(r)} = \frac{\partial}{\partial m(r)} \int \left[g^2(r') - 2 g(r') \int K(r, r') m(r) dr + \left(\int K(r, r') m(r) dr \right)^2 \right] dr'$$

se observa que el primer término, que involucra sólo a $g(r')$, se anula. Con ésto, queda

$$\frac{\partial F}{\partial m(r)} = -2 \int g(r') \int K(r, r') dr dr' + 2 \int \left[\int K(\tilde{r}, r') m(\tilde{r}) d\tilde{r} \int K(r, r') dr \right] dr'$$

Reordenando los integrandos y el orden de integración, tenemos

$$\begin{aligned} \frac{\partial F}{\partial m(r)} = & -2 \int \left[\int g(r') K(r, r') dr' \right] dr + \\ & 2 \int \left[\int m(\tilde{r}) \int K(\tilde{r}, r') K(r, r') dr' d\tilde{r} \right] dr \end{aligned} \quad (63)$$

Ahora resolveremos el segundo término de la ec.(61). Como el primer término del funcional F no depende de la función m_x , sólo se resuelve la expresión

$$\frac{\partial}{\partial x} \frac{\partial F}{\partial m_x} = \frac{\partial}{\partial x} \frac{\partial}{\partial m_x} \alpha \int [m_x^2 + m_y^2] dr$$

realizando la derivación, la expresión resultante queda

$$\frac{\partial}{\partial x} \frac{\partial F}{\partial m_x} = 2\alpha \int m_{xx} dr \quad (64)$$

donde $m_{xx} = \partial^2 m(r) / \partial x^2$.

De la misma forma se resuelve el tercer término de la ec.(61)

$$\frac{\partial}{\partial y} \frac{\partial F}{\partial m_y} = \frac{\partial}{\partial y} \frac{\partial}{\partial m_y} \alpha \int [m_x^2 + m_y^2] dr$$

teniendo como resultado

$$\frac{\partial}{\partial y} \frac{\partial F}{\partial m_y} = 2\alpha \int m_{yy} dr \quad (65)$$

Sustituyendo las expresiones (63), (64) y (65) en la expresión (61), tenemos

$$\begin{aligned} -2 \int \left[\int g(r') K(r, r') dr' \right] dr + 2 \int \left[\int m(\tilde{r}) \int K(\tilde{r}, r') K(r, r') dr' d\tilde{r} \right] dr - \\ 2\alpha \int m_{xx} dr - 2\alpha \int m_{yy} dr = 0 \end{aligned}$$

agrupando y eliminando términos

$$\int \left[\int m(\tilde{r}) \int K(\tilde{r}, r') K(r, r') dr' d\tilde{r} - \int g(r') K(r, r') dr' - \alpha \nabla^2 m(r) \right] dr = 0$$

donde $\nabla^2 m(r) = m_{xx} + m_{yy}$ es el Laplaciano. La igualdad anterior se cumple sólo si el

integrando se hace cero, esto es

$$\int m(\tilde{r}) \int K(\tilde{r}, r') K(r, r') dr' d\tilde{r} - \int g(r') K(r, r') dr' - \alpha \nabla^2 m(r) = 0$$

reordenando los términos de esta expresión, se observa que la función $m(r)$ debe satisfacer la ecuación

$$\alpha \nabla^2 m(r) = \int K(r, r') \left[\int K(\tilde{r}, r') m(\tilde{r}) d\tilde{r} - g(r') \right] dr' \quad (66)$$

Para el caso de que la variable r' sea discreta (r_i) y de que se tenga una estimación del error de las observaciones, esta expresión puede reescribirse en términos de operadores en variable discreta

$$\alpha \nabla^2 m(r) = K^T(r) C_{yy}^{-1} (K m - g)$$

donde C_{yy}^{-1} es la matriz de covarianza de los errores de las observaciones, y

$$(K m)_i = \int K_i(r) m(r) dr$$

