

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE INGENIERIA MECANICA
Y ELECTRICA

Escuela de Graduados



REGRESION NO-LINEAL
UN ENFOQUE PRACTICO

T E S I S

QUE PARA OBTENER EL TITULO DE
MAESTRIA EN CIENCIAS DE LA ADMINISTRACION
ESPECIALIDAD: SISTEMAS

PRESENTA:
JUAN MOISES ARIAS NIEVES

MONTERREY, N. L.

AGOSTO DE 1986

TM

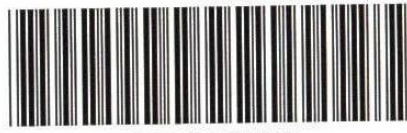
Z5853

.M2

FIME

1986

A7



1020070582

TM
8 2

·M
FIME

/



137783

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE INGENIERIA MECANICA
Y ELECTRICA

Escuela de Graduados



REGRESION NO-LINEAL
UN ENFOQUE PRACTICO

T E S I S

QUE PARA OBTENER EL TITULO DE
MAESTRIA EN CIENCIAS DE LA ADMINISTRACION
ESPECIALIDAD: SISTEMAS

PRESENTA:
JUAN MOISES ARIAS NIEVES

MONTERREY, N. L.

PROLOGO.

Una idea motivante que influye en la intención de elaborar la presente tesis surge de la importancia que tanto teórica como prácticamente tienen los métodos de la regresión no lineal en una amplia diversidad de campos de estudio y de investigación, que tan solo por mencionar algunos, señalamos a la Teoría del Crecimiento y el Análisis de las Series de Tiempo. Aunque no es nuestro objetivo, el de exponer aplicación particular de la regresión no lineal, se quiere, no dejar de señalar tal importancia.

Por otro lado, hacemos saber que se ha desarrollado el presente trabajo teniendo en mente el siguiente doble propósito: Primeramente, se persigue el objetivo de exponer conceptualmente tres métodos comúnmente utilizados en la regresión no lineal o también llamada minimización de una función objetivo suma de cuadrados, seguidamente se tiene el propósito de presentar las rutinas computacionales correspondientes a los algoritmos de los

métodos presentados; son estos los objetivos que se han preten__
dido alcanzar una vez que se decide dar inicio a la elaboración
de las presentes notas.

La intención de presentar conjuntamente los métodos de regre__
sión con su respectiva rutina computacional obedece principalmen__
te al hecho de que tales métodos son iterativos y consecuente__
mente la solución de un problema práctico requiere de resolverlo
computacionalmente, de tal manera que se hace necesario hacer la
presentación conjunta, del método con su rutina computacional que
le corresponde, para lograr en el presente trabajo tener una ex__
posición completa de los métodos de la regresión no-lineal. El
lado débil de la anterior intención, lo constituye el hecho de
que, las rutinas computacionales día a día son mejoradas, no
obstante tal circunstancia, consideramos incompleta, la presenta__
ción únicamente del método, teniendo en cuenta el objetivo prin__
cipal del presente trabajo.

Adicionalmente señalamos, que se tomó en cuenta el deseo de
satisfacer el interés de algún lector que pretenda profundizar
o ampliar alguno de los temas que aquí se abordan, intentamos
para ello, proporcionar una pequeña recopilación bibliográfica
llevada a cabo durante la elaboración del trabajo que nos ocupó.

Finalmente señalamos que el contenido de nuestro trabajo se
vera altamente influenciado por la escasa disponibilidad de
material bibliográfico con que se contó para su elaboración,
situación totalmente ajena a nuestros deseos. Después de este
intento de prólogo no resta más que agradecer a todos los com__
pañeros maestros y alumnos de la Escuela de Graduados de la
Facultad de Ingeniería Mecánica y Eléctrica de la Universidad

Autónoma de Nuevo León, que con su ayuda hicieron posible la aparición de las actuales notas y muy especialmente al Ing. Victoriano Alatorre asesor de la tesis, por sus valiosas re__comendaciones y su apoyo bibliográfico, así mismo como al per__sonal del Centro de Informática de la Facultad de Contaduría Pública y Administración de la U.A.N.L. por facilitar el acce__so al equipo computacional, a todos ellos nuestro más sincero agradecimiento.

J. Moisés Arias N.

Marzo de 1986.

Indice de Secciones

Sección	Página
I.- Introducción	1
II.- Mínimos cuadrados en el caso no-lineal	6
III.- Método de Gauss-Newton	10
IV.- Descripción del programa para el Método de Gauss-Newton	20
V.- Programa para el Método de Gauss-Newton	23
VI.- Interpretación geométrica del Método de Gauss-Newton	34
VII.- Método del Compromiso de Marquardt	40
VIII.- Descripción del programa para el Método de Marquardt	50
IX.- Programa para el Método de Marquardt	52
X.- Apendice A Método de Powell	59
XI.- Apendice B Descripción del programa para el Método de Powell	63
XII.- Apendice C Programa para el Método de Powell	65
Conclusiones	79
Bibliografía	82

I.- INTRODUCCION.

En los cursos elementales de Estadística, los modelos de regresión, muestran "linealidad" en los parámetros y son los modelos del siguiente tipo:

$$Y = \beta_0 + \sum_{i=1}^p \beta_i Z_i (X_1, X_2, \dots, X_k) + \varepsilon \quad (1)$$

donde $Z_i(X_1, X_2, X_3, \dots, X_k)$ representa cualquier función de las variables independientes básicas $X_1, X_2, X_3, \dots, X_k$, este modelo también es conocido como modelo de regresión lineal múltiple.

La ecuación (1) representa una amplia variedad de relaciones, pero existen situaciones en las cuales los modelos de la anterior forma no son los apropiados para ser utilizados.

En algunos casos se puede disponer de información acerca de la relación existente entre las variables independientes

$X_1, X_2, X_3, \dots, X_k$ y la variable dependiente Y , que hacen concluir la necesidad de utilizar un modelo matemático diferente al presentado anteriormente, en algunos otros casos la información de que se dispone puede dejar a disposición, alternativas de varios modelos y aunque pudiera resultar más cómodo la utilización de un modelo lineal, pudiera ser una alternativa menos realista que la opción de utilizar un modelo no-lineal.

Primeramente definimos que cualquier modelo que no pueda ser representado bajo transformaciones algebraicas a uno representado por la ecuación (1), le llamamos modelo no-lineal. Así por ejemplo

$$Y = \left[\theta_1 / (\theta_1 - \theta_2) \right] (e^{-\theta_2 t} - e^{-\theta_1 t}) + \epsilon$$

sera considerado como un modelo no-lineal. Pero no es el caso del modelo

$$Y = \text{EXP}(\theta_1 + \theta_2 t^2 + \epsilon)$$

puesto que puede ser transformado a

$$Y' = \theta_1 + \theta_2 t^2 + \epsilon$$

donde $Y' = \ln(Y)$; mostrando "linealidad" en los parámetros θ_1, θ_2 .

Otros ejemplos de modelos no-lineales se pueden encontrar en diversas áreas, como casos señalamos brevemente los siguientes: (ver referencia (4)).

$$Y = \theta_1 + \theta_2 \text{EXP}(\theta_3 t) + \epsilon$$

Al anterior modelo se le conoce como Ley de Mitscherlich y en química a la curva que tiene como gráfica este modelo

se le designa como curva de reacción de primer orden.

Algunos otros modelos no-lineales están relacionados con conductas de crecimiento, estos modelos, tienen aplicación en una gran variedad de campos, como son: Biología, Ecología, Ciencias Políticas, Ciencias Económicas, Demografía, etc. El tipo de modelo que se necesite en un estudio o investigación, depende del tipo de crecimiento que ocurre, algunos de estos, caen, dentro de la clasificación de modelos de crecimiento mecanicistas y la otra clasificación son los llamados modelos empíricos. Un modelo mecanicista, usualmente, es el resultado de hacer supuestos acerca del tipo de crecimiento, estableciendo a menudo ecuaciones diferenciales que representan estos supuestos y se procede a resolver las ecuaciones para la obtención del modelo requerido. Por otro lado, un modelo empírico, es un modelo que es seleccionado, como su nombre lo indica, empíricamente, aproximado a un modelo mecanicista: típicamente, un modelo empírico es ajustado a un polinomio de orden accesible.

A continuación señalamos, como ejemplo, un modelo de crecimiento

$$Y = \alpha (1 - e^{-kt}) + \varepsilon$$

particularmente a esta ecuación se le conoce con el nombre de " función de crecimiento monomolecular ".

Adicionalmente, señalamos el modelo de Von Bertalanffy (iniciador de la Teoría de Sistemas), modelo de cuatro parámetros y que tiene la forma

$$Y = \left[\theta_1^{1+\theta_2} + \theta_3 e^{\theta_4 t} \right] \frac{1}{1 + \theta_2}$$

donde $\theta_1, \theta_2, \theta_3, \theta_4$ son los parámetros que han de ser estimados.

Otro ejemplo de un modelo no-lineal, es el caso de " Ley de Crecimiento Logístico" , curva que ha desempeñado un importante papel en los estudios del crecimiento de poblaciones humanas. Esta curva da muy buen ajuste al crecimiento de la población en los Estados Unidos de América, de acuerdo con los resultados de los censos poblacionales y limitado a un cierto intervalo de tiempo, el modelo en cuestión es

$$Y = \theta_1 / (1 + \theta_2 \rho^t)$$

Finalmente y de manera muy breve, señalamos otro importante campo de aplicación de los modelos no-lineales; existen ciertos modelos de Series de Tiempo que requieren un tratamiento con métodos de regresión no-lineal (ver referencia (5)) en particular son los modelos de Box y Jenkins. Puesto que un enfoque usual en los modelos y particularmente en los métodos de la regresión no-lineal son los procesos iterativos, es así que, entonces los algoritmos requieren de estimaciones iniciales, ("estimaciones preliminares" como le llaman Box y Jenkins), de tal manera que los autores mencionados han propuesto un método, en el cual, estas estimaciones son obtenidas a través de relaciones que ligan los parámetros involucrados en el modelo de la serie de tiempo, con las autocorrelaciones.

Hemos expuesto hasta aquí, de manera muy breve, alguna de las aplicaciones de los modelos de regresión no-lineal, con

la intención de resaltar la importancia práctica de tales modelos y aunque no es el objetivo del presente trabajo de detallar aplicación particular alguna, si se consideró conveniente mencionarlas aunque fuese meramente de manera superficial.

II.- MINIMOS CUADRADOS EN EL CASO NO-LINEAL.

II.- MINIMOS CUADRADOS EN EL CASO NO-LINEAL.

En el presente trabajo vamos a intentar utilizar una notación lo más estandar posible, iniciamos con la forma del modelo postulado

$$Y = f(X_1, X_2, \dots, X_k : \theta_1, \theta_2, \dots, \theta_p) + \epsilon \quad (2)$$

donde Y es la variable dependiente : X_i ($i=1,2, \dots, k$) son las variables independientes y θ_j ($j=1,2, \dots, p$) son los parámetros presentes en el modelo.

Antes de continuar, haremos la siguiente observación: La notación vectorial o matricial que empleamos , es la de subrayar el nombre de la variable que represente a un vector o la variable que representa a una matriz, así por ejemplo

$$\underline{X} = \begin{pmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_k \end{pmatrix} = (X_1, X_2, \dots, X_k)^T$$

$$\underline{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix} = (\theta_1, \theta_2, \dots, \theta_p)^T$$

donde el superíndice T, significa transposición. De manera que la ecuación (2) queda representada como

$$\underline{y} = f(\underline{x}; \underline{\theta}) + \varepsilon$$

Suponemos que $E(\varepsilon) = 0$; $\text{Var}(\varepsilon) = \sigma^2$ y que los errores son independientes y no correlacionados además aditivos es decir que en nuestro modelo los errores estarán siempre sumandose.

Denotamos las n observaciones de la forma siguiente

$$y_u, x_{1u}, x_{2u}, x_{3u}, \dots, x_{ku}$$

para $u=1,2,3, \dots, n$

Entonces alternativamente, nuestro modelo postulado toma la forma

$$y_u = f(x_u; \underline{\theta}) + \varepsilon_u \quad (3)$$

donde

$$\underline{x}_u = (x_{1u}, x_{2u}, x_{3u}, \dots, x_{ku})^T$$

adicionalmente suponemos

$$\underline{\varepsilon} \sim N(\underline{0}_{n \times n}; \underline{I}_{n \times n} \sigma^2)$$

donde

$$\underline{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T$$

$\underline{0}_{n \times n}$ denota la matriz cero de orden $n \times n$.

$\underline{I}_{n \times n}$ denota la matriz identidad de orden $n \times n$.

Ahora definimos la suma de los errores al cuadrado del modelo no-lineal como la función

$$S(\underline{\theta}) = \sum_{u=1}^n \left[y_u - f(\underline{x}_u; \underline{\theta}) \right]^2 \quad (4)$$

observemos que $S(\underline{\theta})$ es una función exclusivamente de $\underline{\theta}$; puesto que y_u , \underline{x}_u son observaciones fijas. Ahora denotamos como $\hat{\underline{\theta}}$ a el estimador mínimo cuadrado de $\underline{\theta}$, esto es, son los valores de $\underline{\theta}$ que minimizan a $S(\underline{\theta})$. Adicionalmente se puede demostrar que bajo el supuesto de normalidad en los errores, es decir bajo el supuesto de $\underline{\varepsilon} \sim N(\underline{0}, I\sigma^2)$ el estimador mínimo cuadrado de $\underline{\theta}$ es también un estimador máximo verosímil de $\underline{\theta}$.

Para encontrar el estimador mínimo cuadrado de $\underline{\theta}$, necesitamos diferenciar la ecuación (4) respecto a $\underline{\theta}$ y obtener las p -ecuaciones normales siguientes

$$\sum_{u=1}^n \left[y_u - f(\underline{x}_u; \hat{\underline{\theta}}) \right] \left[\frac{\partial f(\underline{x}_u; \underline{\theta})}{\partial \theta_i} \right]_{\underline{\theta} = \hat{\underline{\theta}}} = 0$$

para $i=1, 2, 3, \dots, p$

Para el caso de regresión lineal múltiple la función $f(\underline{x}_u; \underline{\theta})$ es lineal y depende únicamente de \underline{x}_u de tal manera que

$$\frac{\partial f}{\partial \theta_i} = x_i \quad \text{para } i=1, 2, \dots, p$$

Así que las ecuaciones normales resultan estar en forma lineal en θ_i , de modo que obtenemos un sistema de ecuaciones lineales

de orden $p \times p$ y que procediendo a resolverlo encontramos el estimador $\hat{\theta}$, pero en el caso que nos ocupa, resolver el sistema que se plantea en la última ecuación, en general no es sencillo entonces se tienen que buscar métodos más especiales, tales métodos están caracterizados por ser aproximados y además iterativos, en situaciones de mayor complejidad, puede ser que la solución no sea única, es decir que el sistema tenga soluciones múltiples, lo que hace que la resolución del mismo sea todavía más complicada. En el presente trabajo intentamos abordar algunos de los métodos que comúnmente se utilizan, pero primeramente exponemos las ideas centrales de los mismos y posteriormente presentamos las respectivas implementaciones algorítmicas en rutinas computacionales, por lo pronto y para finalizar la actual sección, solamente mencionamos los nombres de los dos métodos que principalmente nos interesa exponer:

- a) Método de "linealización" o de Gauss-Newton.
- b) Método del Compromiso de Marquardt.

Adicionalmente, de manera esquemática y a manera de apéndice presentamos:

- c) Método de las Diferencias Finitas de Powell.

III.- METODO DE GAUSS-NEWTON.

III.- METODO DE GAUSS-NEWTON.

Este método utiliza los resultados de mínimos cuadrados lineales en un proceso iterativo, es decir en etapas sucesivas. Suponemos primeramente que el modelo postulado es de la forma presentada en la ecuación (3). Sea θ_{i0} ; $i=1,2, \dots, p$ los valores iniciales de los parámetros involucrados en el modelo; estos valores pueden ser obtenidos por supuestos o estimaciones preliminares, basadas en la información con la que se cuente o también pueden ser obtenidos subjetivamente, tomando en cuenta las propuestas de los investigadores con experiencia y conocimiento del fenómeno o problema en cuestión. Para el método de Gauss-Newton es de gran importancia los valores iniciales puesto que una buena solución inicial proporciona una rápida convergencia; finalmente, respecto a la cuestión de los valores de inicio diremos que existen algunas recomendaciones útiles que son posi...

bles de tomar en cuenta para integrar un buen conjunto de va lores iniciales, pero esto puede constituir la base o la idea central de un trabajo o estudio particular, que por razones, tanto de tiempo como de objetivos, no es posible abordar por ahora.

Iniciamos con el desarrollo en una serie de Taylor, para la función $f(\underline{X}_u; \underline{\theta})$ alrededor de $\underline{\theta} = \underline{\theta}_0 = (\theta_{10}, \theta_{20}, \dots, \theta_{p0})^T$ y truncamos el desarrollo hasta las primeras derivadas con la intención de eliminar los términos no-lineales y así obtenemos la siguiente aproximación cuando $\underline{\theta}$ está cercano a $\underline{\theta}_0$:

$$f(\underline{X}_u; \underline{\theta}) \doteq f(\underline{X}_u; \underline{\theta}_0) + \sum_{i=1}^p \left[\frac{\partial f(\underline{X}_u; \underline{\theta})}{\partial \theta_i} \right]_{\underline{\theta} = \underline{\theta}_0} (\theta_i - \theta_{i0})$$

Ahora haremos

$$f_u^0 = f(\underline{X}_u; \underline{\theta}_0)$$

$$\beta_i^0 = \theta_i - \theta_{i0}$$

$$z_{iu}^0 = \left[\frac{\partial f(\underline{X}_u; \underline{\theta})}{\partial \theta_i} \right]_{\underline{\theta} = \underline{\theta}_0}$$

Observemos entonces que la ecuación (3) toma entonces la forma siguiente

$$y_u - f_u^0 = \sum_{i=1}^p \beta_i^0 z_{iu}^0 + \varepsilon_u \quad (5)$$

observamos que entonces esta última ecuación es de la forma mostrada en la ecuación (1). Estimemos ahora los parámetros β_i^0 ; $i=1,2, \dots, p$ aplicando la teoría de mínimos cuadrados lineales, pero antes vamos a utilizar la siguientes notación matricial.

$$\underline{z}_0 = \begin{pmatrix} z_{11}^0 & z_{21}^0 & z_{31}^0 & \dots & z_{p1}^0 \\ z_{12}^0 & z_{22}^0 & z_{32}^0 & \dots & z_{p2}^0 \\ z_{13}^0 & z_{23}^0 & z_{33}^0 & \dots & z_{p3}^0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ z_{1n}^0 & z_{2n}^0 & z_{3n}^0 & \dots & z_{pn}^0 \end{pmatrix} = \begin{bmatrix} z_{iu}^0 \end{bmatrix}_{n \times p}$$

$$\underline{y}_0 = \underline{y} - \underline{f}^0 = \begin{pmatrix} y_1 - f_1^0 \\ y_2 - f_2^0 \\ y_3 - f_3^0 \\ \cdot \\ \cdot \\ \cdot \\ y_n - f_n^0 \end{pmatrix}$$

$$\underline{b}_0 = \begin{pmatrix} b_1^0 \\ b_2^0 \\ b_3^0 \\ \cdot \\ \cdot \\ \cdot \\ b_p^0 \end{pmatrix}$$

$$\underline{\beta}_0 = \begin{pmatrix} \beta_1^0 \\ \beta_2^0 \\ \beta_3^0 \\ \cdot \\ \cdot \\ \cdot \\ \beta_p^0 \end{pmatrix}$$

Entonces la ecuación escalar representada por (5) queda representada vectorialmente por

$$\underline{y}_0 = \underline{y} - \underline{f}^0 = \underline{z}_0 \underline{\beta}_0 + \underline{\varepsilon} \quad (6)$$

donde $\underline{\varepsilon}$ es un vector $n \times 1$ de errores.

Así que la suma de los errores al cuadrado es

$$\begin{aligned} \underline{\varepsilon}^T \underline{\varepsilon} &= \left[(\underline{y} - \underline{f}^0) - \underline{z}_0 \underline{\beta}_0 \right]^T \left[(\underline{y} - \underline{f}^0) - \underline{z}_0 \underline{\beta}_0 \right] \\ &= \left[(\underline{y} - \underline{f}^0)^T - \underline{\beta}_0^T \underline{z}_0^T \right] \left[(\underline{y} - \underline{f}^0) - \underline{z}_0 \underline{\beta}_0 \right] \\ &= (\underline{y} - \underline{f}^0)^T (\underline{y} - \underline{f}^0) - \underline{\beta}_0^T \underline{z}_0^T (\underline{y} - \underline{f}^0) - (\underline{y} - \underline{f}^0)^T \underline{z}_0 \underline{\beta}_0 + \underline{\beta}_0^T \underline{z}_0^T \underline{z}_0 \underline{\beta}_0 \end{aligned}$$

observamos que $\underline{\beta}_0^T \underline{z}_0^T (\underline{y} - \underline{f}^0)$ es una cantidad escalar, puesto que

$$\left[\underline{\beta}_0^T \right]_{1 \times p} \left[\underline{z}_0^T \right]_{p \times n} \left[\underline{y} - \underline{f}^0 \right]_{n \times 1}$$

entonces, también la matriz transpuesta es una cantidad escalar

$$\left[\underline{\beta}_0^T \quad \underline{z}_0^T (\underline{y} - \underline{f}^0) \right]^T = (\underline{y} - \underline{f}^0)^T \underline{z}_0 \underline{\beta}_0$$

así que

$$\underline{\varepsilon}^T \underline{\varepsilon} = (\underline{y} - \underline{f}^0)^T (\underline{y} - \underline{f}^0) - 2(\underline{y} - \underline{f}^0)^T \underline{z}_0 \underline{\beta}_0 + \underline{\beta}_0^T \underline{z}_0^T \underline{z}_0 \underline{\beta}_0 \quad (7)$$

Vamos ahora a diferenciar esta última ecuación (7) con respecto a $\underline{\beta}_0$ e igualarla a cero y al mismo tiempo reemplazar $\underline{\beta}_0$ por \underline{b}_0 el cuál, es el estimador mínimo cuadrado de $\underline{\beta}_0$

Observación: Para obtener la derivada de $\underline{\varepsilon}^T \underline{\varepsilon}$ respecto a $\underline{\beta}_0$, diferenciamos respecto a cada componente de $\underline{\beta}_0$ y formamos una matriz de orden $p \times 1$, observamos además que $\underline{z}_0^T \underline{z}_0$ es una matriz

I.- INTRODUCCION.

simétrica, entonces

$$\begin{aligned} \frac{\partial}{\partial \beta_0} (\underline{\varepsilon}^T \underline{\varepsilon}) &= \frac{\partial}{\partial \beta_0} (\underline{Y} - \underline{f}_0)^T (\underline{Y} - \underline{f}_0) - 2 \frac{\partial}{\partial \beta_0} [\beta_0^T \underline{Z}_0^T (\underline{Y} - \underline{f}_0)] + \frac{\partial}{\partial \beta_0} [\beta_0^T \underline{Z}_0^T \underline{Z}_0 \beta_0] = 0 \\ &= -2 \underline{Z}_0^T (\underline{Y} - \underline{f}_0) + 2 \left[\underline{Z}_0^T \underline{Z}_0 \beta_0 \right]_{\beta_0 = \underline{b}_0} = 0 \end{aligned}$$

Entonces

$$\underline{Z}_0^T \underline{Z}_0 \underline{b}_0 = \underline{Z}_0^T (\underline{Y} - \underline{f}_0)$$

$$\underline{b}_0 = (\underline{Z}_0^T \underline{Z}_0)^{-1} \underline{Z}_0^T (\underline{Y} - \underline{f}_0) \quad (8)$$

por tanto el vector \underline{b}_0 minimiza la suma de cuadrados

$$S(\underline{\theta}) = \sum_{u=1}^n \left[Y_u - f(X_u; \underline{\theta}) - \sum_{i=1}^p \beta_i Z_{iu} \right]^2$$

con respecto a β_i^0 ; $i=1,2,3, \dots, p$ donde

$$\beta_i^0 = \theta_{i1} - \theta_{i0}$$

entonces

$$b_1^0 = \theta_{11} - \theta_{10}$$

así que

$$\theta_{i1} = b_i^0 + \theta_{i0}$$

donde θ_{i1} es el estimador revisado de θ_i

Es así que podemos colocar el valor de θ_{i1} , el estimador revisado, en el mismo papel jugado por los valores de θ_{i0} e iniciar nuevamente el procedimiento desarrollado, pero reemplazando todo subíndice cero, por uno.

Vamos a expresar los resultados anteriores pero de manera vectorial

$$\underline{\theta}_{j+1} = \underline{\theta}_j + \underline{b}_j$$

$$\underline{\theta}_{j+1} = \underline{\theta}_j + \left(\underline{Z}_j^T \underline{Z}_j \right)^{-1} \underline{Z}_j^T (\underline{Y} - \underline{f}^j)$$

donde

$$\underline{Z}_j = \begin{matrix} z_{1j}^j \\ z_{2j}^j \\ \vdots \\ z_{pj}^j \end{matrix}$$

$$\underline{f}^j = (f_1^j, f_2^j, \dots, f_n^j)^T$$

$$\underline{\theta}_j = (\theta_{1j}, \theta_{2j}, \dots, \theta_{pj})^T$$

De esta manera nuestro proceso es iterativo; interrumpiendolo hasta que se alcance la convergencia deseada. Una prueba de convergencia que se propone es, detener el proceso en el momento en que, las iteraciones sucesivas j y $j+1$ sean tales que

$$\left| \frac{\theta_{i(j+1)} - \theta_{ij}}{\theta_{ij}} \right| \leq \delta \quad \text{para } i=1,2,3, \dots, p$$

donde δ es una cantidad pequeña establecida previamente.

Resolvemos ahora un ejemplo para ilustrar numéricamente el método de Gauss-Newton, pero antes se hacen las siguientes observaciones:

- El ejemplo presentado, es el mismo problema que se resuelve en el programa computacional que más adelante se presenta. Las diferencias que surgen entre la solución a el ejemplo y la salida de resultados del programa son debidas principalmente a la utilización de coeficientes de penalidad en el programa mencionado, pero tales diferencias no deben de ser significativamente diferentes, otro factor que contribuye, es el debido a errores de redondeo.

- En el ejemplo que a continuación se presenta, todos los cálculos fueron obtenidos mediante la elaboración de un pequeño programa, utilizando el lenguaje computacional BASIC, por lo cómodo en el manejo de arreglos bidimensionales.

Nuestro modelo es:

$$Y = f(X_1; \theta_1, \theta_2, \theta_3) = \theta_1 + \theta_2 \text{EXP}(\theta_3 X_1)$$

donde las observaciones son

u	Y_u	X_{1u}
1	127	- 5
2	151	- 3
3	379	- 1
4	421	+ 1
5	460	+ 3
6	426	+ 5

Obtendremos primeramente

$$\left[Z_{iu} \right] = \left[\frac{\partial f(\underline{X}; \underline{\theta})}{\partial \theta_i} \right]$$

evaluandolas en $\underline{\theta} = \underline{\theta}_0$ donde $\underline{\theta}_0$ tiene los siguientes valores iniciales :

$$\underline{\theta}_0 = \begin{pmatrix} \theta_{10} = 500 \\ \theta_{20} = -150 \\ \theta_{30} = -0.2 \end{pmatrix}$$

entonces

$$\begin{aligned} \left[Z_{iu} \right]_{6 \times 3} &= \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \theta_3} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \frac{\partial f_2}{\partial \theta_3} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_6}{\partial \theta_1} & \frac{\partial f_6}{\partial \theta_2} & \frac{\partial f_6}{\partial \theta_3} \end{pmatrix} \\ &= \begin{pmatrix} 1 & e^{\theta_3 \times 11} & \theta_2 \times 11 e^{\theta_3 \times 11} \\ 1 & e^{\theta_3 \times 12} & \theta_2 \times 12 e^{\theta_3 \times 12} \\ \vdots & \vdots & \vdots \\ 1 & e^{\theta_3 \times 16} & \theta_2 \times 16 e^{\theta_3 \times 16} \end{pmatrix} \end{aligned}$$

evaluandola en $\underline{\theta}_0$

$$\begin{bmatrix} z_{iu}^0 \end{bmatrix}_{6 \times 3} = \begin{pmatrix} 1 & 2.7182 & -13.5914 \\ 1 & 1.822 & -5.4663 \\ 1 & 1.2214 & -1.2214 \\ 1 & 0.8187 & 0.8187 \\ 1 & 0.5488 & 1.6464 \\ 1 & 0.3679 & 1.8394 \end{pmatrix}$$

hacemos $\underline{z}^0 = \begin{bmatrix} z_{iu} \end{bmatrix}_{6 \times 3}$

así que

$$\underline{z}_0^T = \begin{pmatrix} 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 2.7182 & 1.822 & 1.2214 & 0.8187 & 0.5488 & 0.3679 \\ -13.5914 & -5.4663 & -1.2214 & 0.8187 & 1.6464 & 1.8394 \end{pmatrix}$$

$$\underline{z}_0^T \underline{z}_0 = \begin{pmatrix} 6.0000 & 7.497 & -15.9746 \\ 7.497 & 13.3069 & -46.1450 \\ -15.9746 & -46.145 & 222.8627 \end{pmatrix}$$

puesto que

$$\underline{y} = \begin{pmatrix} 127 \\ 151 \\ 379 \\ 421 \\ 460 \\ 426 \end{pmatrix} \quad \underline{f}^0 = \begin{pmatrix} 92.2577 \\ 226.6822 \\ 316.7896 \\ 377.1904 \\ 417.6783 \\ 444.8181 \end{pmatrix}$$

entonces

$$(\underline{Z}_0^T \underline{Z}_0)^{-1} \underline{Z}_0^T (\underline{Y} - \underline{f}^0) = \begin{pmatrix} 23.4008 \\ -7.06103 \\ -0.0698156 \end{pmatrix}$$

por tanto

$$\underline{B}_1 = \begin{pmatrix} 500 \\ -150 \\ -0.20 \end{pmatrix} + \begin{pmatrix} 23.4008 \\ -7.06103 \\ -0.0698156 \end{pmatrix} = \begin{pmatrix} 523.4008 \\ 157.06103 \\ -0.2698156 \end{pmatrix}$$

Finalmente aplicamos una prueba de convergencia, si esta es satisfecha entonces detenemos el proceso, es decir no se realizan más iteraciones, de otra manera, iniciamos el proceso nuevamente.

IV.- DESCRIPCION DEL PROGRAMA PARA EL METODO
DE GAUSS-NEWTON.

IV.- DESCRIPCION DEL PROGRAMA PARA EL METODO DE
DE GAUSS NEWTON.

Todos los programas que se presentan en este trabajo, fueron tomados y modificados apropiadamente de los que aparecen publicados en la obra:

Optimization techniques with FORTRAN

J. L. Kuester and J. H. Mize

Mc. Graw-Hill Book Company, 1973

A su vez el programa que los autores presentan en la anterior obra, lo basaron en:

Nonlinear Parameters Estimation and Programming

Catalog of Programs for IBM System 360

Models 25 and Above, 20-1619-8

Program number 360.D-13.6.003

International Business Machines Corp.

El programa consta de un programa principal y seis subrutinas en las cuales, la transferencia de datos se realiza mediante la instrucción COMMON, del lenguaje FORTRAN y siendo el propósito de cada una de las subrutinas los siguientes:

1.-El programa principal llamado START-GAUSS-METHOD define canales de entrada/salida, haciendo un solo llamado a la subrutina principal.

2.- La subrutina DLSQ, se utiliza para el cálculo de derivadas.

3.- La subrutina NLMAX, es la subrutina principal y su propósito es el de llevar el control de las demás subrutinas coordinando todos los cálculos.

4.- La subrutina ACCUM, calcula los valores de la función mínimos cuadrados.

5.- La subrutina OUT, es el control de las impresiones.

6.- La función FUNC, especifica el modelo.

7.- La subrutina INVER, es utilizada para calcular matrices inversas.

Vamos ahora a dar la descripción de algunos parámetros utilizados en el programa:

NTH, es el número de parámetros a estimar en el modelo.

LOUT, controlador de impresiones de resultados intermedios.

NPH, controlador de la función de penalidad.

C1, vector para el almacenamiento de valores iniciales.

M, número de observaciones.

NA, total de variables, dependientes e independientes.

A, matriz de valores observados.

CLB, cota inferior en los valores de los parámetros del modelo.

CUB, cota superior en los valores de los parámetros del modelo.

II, define tareas en las subrutinas.

I, es el índice de los puntos de valores observados.

H, valor de la cota mínima.

El modelo experimental, para la corrida es el mismo modelo presentado en el ejemplo al final de la sección "Método de Gauss-Newton". Es el mismo modelo en los otros dos programas, que se presentan posteriormente, siendo la intención, la de comparar los resultados que se obtengan.

V.- PROGRAMA PARA EL METODO DE GAUSS-NEWTON.

```

C
!-----!
! SUBROUTINA INICIAL, DEFINIMOS CANALES DE ENTRADA/SALIDA !
! Y SE HACE UN UNICO LLAMADO A LA SUBROUTINA PRINCIPAL !
!-----!
COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
*,CUB(20),CLB(20),PNL(20),NCON,LQUT,F3,NTH,F6,F7,METH,NPH
*,MD,LS,C1(20)
*,X,XTH(20),A(200,10),M,NA

C
OPEN(UNIT=6,NAME='GAUSSNUM4.DAT',STATUS='NEW')
NO=6

C
WRITE (NO,1)
1 FORMAT(5(/),2X' REGRESION NO-LINEAL POR EL METODO
* DE GAUSS-NEWTON O DE "LINEALIZACION" ')

C
CALL NLMAX(NI,NO)

C
END

!-----!
! EN ESTA SUBROUTINA SE CALCULAN DERIVADAS NUMERICAMENTE !
!-----!
SUBROUTINE ELSQ(I1,I)
COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
*,CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NPH
*,MD,LS,C1(20)
*,X,XTH(20),A(200,10),M,NA
GO TO (1,1,2),I1
2 RETURN
1 X=FUNC(C1,A,I)
GO TO (5,6),I1
6 DO 10 J=1,NTH
C1(J)=C1(J)+0.0001*C1(J)
FOR=FUNC(C1,A,I)
C1(J)=C1(J)-0.0002*C1(J)
REV=FUNC(C1,A,I)
C1(J)=C1(J)+0.0001*C1(J)
XTH(J)=(FOR-REV)/(0.0002*C1(J))
10 CCNTINUE
C
5 RETURN
END

!-----!
! SUBROUTINA PRINCIPAL,ES DONDE LLEVAMOS CONTROL DE LAS !
! DEMAS SUBROUTINA. !
!-----!
SUBROUTINE NLMAX(NI,NO)
C--- DECK-1 METCDC GAUSS-NEWTON
C
EQUIVALENCE(NTH,L)
COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
*,CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NPH
*,MD,LS,C1(20)
*,X,XTH(20),A(200,10),M,NA
*,XTH(20,20)
*,DIF(100)
C--- SE AGREGO DIMENSION W()
C

```

DIMENSION W(20)
METH=1

24

```
C
DATA NTH,LOUT,MD,M/3,1,1,6/
IF(NTH.EQ.0)CALL EXIT
DATA C1(1),C1(2),C1(3)/500.00,-150.00,-0.20/

C
LS=1

C
CALL ACCUM(3,NI,NO)
CALL BOUND(3,H,NI,NO)
WRITE(NO,5000)(C1(I),I=1,L)
5000 FORMAT(//,2X,'PARAMETROS INICIALES:',/,15X,(7E16.6))
IF (LS-3)199,907,199
199 IPH=2
NIN=0
NF=0
ND=0
EPS=1.E-4
EPS1=1.E-3
DO 906 I=1,L
FF(I)=C1(I)
Y(I)= 0.0 - C1(I)
906 CCNTINUE
H=1.0
CALL BCUND(4,F,NI,NO)
DO 911,I=1,L
911 Y(I)=C1(I)*H
NPH=1
IF(NCON)1,899,1
899 NPH=2
GOTO 16

C
1 GOTO (212,16)LOUT
212 WRITE(NO,1001)
1001 FORMAT(//,2X,' INCLUYENDO FUNCION DE PENALIDAD',//)
16 II=2
NRE=1
ND=ND+1
GOTO 100

C
51 II=1
100 NF=NF+1
LS=1
CALL ACCUM(II,NI,NC)
F4=F3
GOTO (405,68,1003)LS
405 GOTO (401,499)NPH
401 CALL BOUND(II,X,NI,NO)
499 GOTO (48,409)II
409 DO 408 I=2,L
DO 408 J=2,I
408 G(I,J-1)=G(J-1,I)
48 GOTO (205,110)LCUT
205 GOTO (208,209)II
209 WRITE(NC,210)ND
210 FORMAT(////,2X,' ITERACION :',I6)
208 WRITE(NO,207)F3,NF,(C1(I),I=1,L)
207 FORMAT(/,74(' '),/,2X,'FUNCION:',E17.7,10X,'EVALUACION:',I6,
*,2X,'VALGR DE LOS PARAMETROS:',/
*,10X,(7E17.7),75(' '),////)
```

```

C
110 GOTO (101,111,894)II
101 IF(F2-F3)22,21,21
22 GOTO (24,16)NPH
24 GOTO (111,16)NRE
111 DO 106 I=1,L
106 FF(I)=C1(I)
IF(11-1)26,26,25
26 NRE=2
GOTO (34,16)NRR
34 Q=0.
IF(NIN)14,14,16
14 CONTINUE
DO 27 I=1,L
27 C=Q+F(I)*Y(I)
H=(Q-2.*(F3-F2))/2./(F3-F2-Q)
IF(ABS(H)-.1)16,16,30
30 IF(H+1.)304,304,31
31 DO 29 I=1,L
29 Y(I)=H*Y(I)
GOTO 304
25 CONTINUE
DO 800 I=1,L
800 DO 800 J=1,L
C(I,J)=G(I,J)
LT3=2
DO 9 I=1,L
EGV(I)=C(I,I)
IF(C(I,I))9,13,10
13 C(I,I)=-1.C
GOTO 9
10 LT3=1
C(I,I)=-C(I,I)
9 CONTINUE
41 CALL INVER(G,L,G1)
DO 11 J=1,L
W(J)=0.0
DC 202 J1=1,M
W(J)=W(J)-XTHT(J,J1)*EIF(J1)
202 CONTINUE
C11 W(J)=W(J)/C(J,J)
11 CONTINUE
DO 201 I=1,L
Y(I)=0.0
DO 201 J=1,L
Y(I)=Y(I)-G1(I,J)*W(J)
201 CONTINUE
304 NRR=1
F2=F3
F6=F7
H=1.
CALL BCUND(4,F,NI,NO)
NIN=NIN/2
H=H/2.**NIN
DO 604 I=1,L
604 Y(I)=H*Y(I)
603 J=1
DO 700 I=1,L
IF(ABS(Y(I))/(EPS1+ABS(FF(I)))-EPS)700,700,701
701 J=2
700 CONTINUE

```

```

GOTO (33,702)J
33  GOTO (898,16)NRE
898  GOTO (1002,897)NPH
897  IF(H-1.)896,1003,1003
896  CALL BOUND(5,H,NI,NO)
      DC 895 I=1,L
895  C1(I)=FF(I)+H*Y(I)
      CALL ACCUM(1,NI,NO)
      NF=NF+1
      II=3
      GOTO (208,894)LOUT
894  IF(F3-F2) 706,1003,1003
702  DO 12 I=1,L
12   C1(I)=FF(I)+Y(I)
      GOTO 51
21   GOTO (2,6)NRE
6    DC 3 I=1,L
3    C1(I)=FF(I)
      GOTO 16
2    Q=0.
      NRR=2
      DO 58 I=1,L
58   Q=Q+F(I)*Y(I)
      H=Q/(Q+(F2-F3))*0.5
66   IF (4.*H-1.)59,59,62
59   H=.25
62   J=1
      NIN=NIN+1
      DO 703 I=1,L
      Y(I)=H*Y(I)
      IF(ABS(Y(I))/(EPS1+ABS(FF(I)))-EPS)703,703,704
704  J=2
703  CONTINUE
      GO TO (705,706),J
705  H=1.
      CALL BOUND(4,H,NI,NO)
      DC 23 I=1,L
      Y(I)=H*Y(I)
23   C1(I)=FF(I)+Y(I)
      GOTO 51
706  F3=F2
      F7=F6
      DC 707 I=1,L
707  C1(I)=FF(I)
19   GOTO (1002,1003)NPH
1002 IF(IPH)1004,1004,1005
1004 NPH=2
      GOTO (121,122)LOUT
C
121  WRITE(NO,214)
214  FORMAT(5X,' SIN FUNCION DE PENALIDAD')
C
122  F2=-1.E30
      GOTO 16 '
1005 IF(ABS(F3-F4)-.1)1004,1006,1006
1006 CALL BOUND(6,H,NI,NO)
      IPH=IPH-1
      GOTO (213,16)LOUT
C
213  WRITE(NO,211)
211  FORMAT(2X,'FUNCION DE PENALIDAD,RECUCIDA FACTOR DE 10')

```



```

GOTO 16
1003 CONTINUE
WRITE(NO,123)F3,(C1(I),I=1,L)
123  FORMAT(/,2X,'MAXIMO DE LA FUNCION OBJETIVO:',E17.7,/,
1 2X,'PARAMETROS:',/, (7E17.7))
WRITE(NO,9001) NF,ND
9001  FORMAT(/,2X,'EVALUACIONES DE LA FUNCION:',I6,
1 /,2X,'EVALUACIONES DE DERIVADAS:',I6)
407  CONTINUE
C
CALL BOUND(7,H,NI,NO)
CALL OUT(NO)
C
GOTO(217,907)LT3
217  WRITE(NO,216)
216  FORMAT(2X,' SOLUCION, NO ES UN MAXIMO INTERIOR')
907  RETURN
68   J=1
DO 71 I=1,L
Y(I)=.5*Y(I)
IF( ABS(Y(I)) / (EPS1+ABS(C1(I))) - EPS)71,71,925
925  J=2
71   C1(I)=C1(I)-Y(I)
GOTO (909,926)J
909  WRITE(NO,910)
910  FORMAT(/,2X,'VALORES FACTIBLES DE LOS PARAMETROS NO PUE
1 DEN SER ENCONTRADOS')
RETURN
926  CONTINUE
WRITE(NG,924)
924  FORMAT(/,2X,'**** VUELVA A INICIAR ****')
C
IF((II-1)51,51,16)
C
END
!-----!
! EN ESTA SUBROUTINA SE VA COMPUTANDO LA FUNCION !
! DE MINIMOS CUADRADOS. !
!-----!
SUBROUTINE ACCUM(II,NI,NO)
COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
*,CUB(20),CLB(20),PNL(20),ACON,LOUT,F3,NTH,F6,F7,METH,NPH
*,MD,LS,C1(20)
*,X,XTH(20),A(200,10),M,NA
*,XTHT(20,20)
*,DIF(100)
C
GOTO (100,100,101)II
C
101  CCNTINUE
DATA A(1,1),A(2,1),A(3,1),A(4,1),A(5,1),A(6,1)
*/-5.0,-3.0,-1.0,1.0,3.0,5.0/
DATA A(1,2),A(2,2),A(3,2),A(4,2),A(5,2),A(6,2)/
*127.0,151.0,379.0,421.0,460.0,426.0/
DATA M,NA/6,2/
C
WRITE(NO,2004)
2004  FORMAT(///,10X,' OBSERVACIONES:',/,3X,'OBS.'
*,5X,' XI ',10X,' YI ')
DO 2005 I=1,M
2005  WRITE(NO,2006) I,(A(I,J), J=1,NA)

```

```

2006  FORMAT(15,7E16.6,/, (E21.6,6E16.6))
      CALL DLSQ(3,0)
      RETURN

```

```

C
100  CCONTINUE
      F3=0.0
      GOTO (1,2)II
      DO 3 I=1,NTH
        F(I)=0.

```

```

C
      GOTO (13,3)METH
13   DO 15 J=1,NTH
15   G(I,J)=0.

```

```

C
      CONTINUE
3    DO 4 MU=1,M
1    CALL DLSQ(II,MU)
      DO 77 J2=1,NTH
        XTHT(J2,MU)=XTH(J2)
77   CONTINUE
      GOTO (6,7)LS
6    CONTINUE
      DIF(MU)=X
      F3=F3-X*X
      GOTO (4,5)II
5    DO 12 I=1,NTH
      F(I)=F(I)-X*XTH(I)

```

```

C
      GOTO (14,12)METH
14   DO 16 J=1,NTH
16   G(I,J)=G(I,J)-XTH(I)*XTH(J)
C
12   CONTINUE
4    CONTINUE
7    CONTINUE
      RETURN
      END

```

```

C
C
!-----!
!   SUBROUTINA PARA IMPRIMIR LOS RESULTADOS FINALES   !
!-----!
      SUBROUTINE CUT (NC)

```

```

C
      COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
      *,CUB(20),CLB(20),PNL(20),ACCN,LOUT,F3,NTH,F6,F7,METH,APH
      *,MD,LS,C1(20)
      *,X,XTH(20),A(200,10),M,NA
      WRITE(NO,3)
3    FORMAT(///,10X,'MODELO',5X,
      * 'Y = A + A * EXP( A * X) ',/,26X,'1',3X,'2',8X,'3',/
      *,//,10X,'RESIDUALES (VALORES COMPUTADOS -
      * VALORES OBSERVADOS)',/)
      J=0
      DO 1 I=1,M
        J=J+1
        CALL DLSQ(1,I)
        F(J)=X

```

```

      IF(J-7)1,2,2
2      J=0
      WRITE(NO,4)(F(K),K=1,7)
4      FORMAT(/,10X,3E16.6,/,10X,3E16.6,/,10X,E16.6,/)
1      CONTINUE
      IF(J)5,6,5
5      WRITE(NO,4)(F(K),K=1,J)
6      F3=-F3
      X=F3/REAL(M-NTH)
      X1=SQRT(X)
      WRITE(NO,7)F3,X1
7      FORMAT(2X,' SUMA DEL CUADRADO DE LOS RESIDUALES:',E17.7,
*///,2X,' DESVIACION ESTANDAR:',16X,2E17.7)
C
C
      DO 302 I=1,NTH
      DO 302 J=1,NTH
302     C(I,J)=X*G(I,J)
      WRITE(NO,28)(C1(I),I=1,NTH)
28     FORMAT(/,2X,' VALOR DE LOS PARAMETROS:',/,
*(15X,7E17.7))
C
      RETURN
      END
!-----!
!   ESTA FUNCION ES DEFINIDA POR EL USUARIO,
!   EN NUESTRO PROGRAMA, LA FUNCION ES:
!
!           ^      ^      ^
!           Y = A  +  A * EXP( A * X)
!              1    2    3
!-----!
C
      FUNCTION FUNC (C1,A,I)
C
C
      DIMENSION C1(20),A(20,10)
      FUNC=C1(1) + C1(2) * EXP(C1(3)*A(I,1)) - A(I,2)
C
      RETURN
      END
!-----!
!   SUBROUTINA PARA CONTROLAR LAS RESTRICCIONES EN LOS VALC--
!   RES DE LOS COEFICIENTES.
!-----!
      SUBROUTINE BCUND (II,H,NI,NO)
C
      COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
*,CUB(20),CLB(20),PNL(20),ACON,LOUT,F3,NTH,F6,F7,METH,NPH
*,MD,LS,C1(20)
C
      GOTO (1,1,2,3,3,44,43)II
C
44     DO 45 I=1,NTH
45     PNL(I)=.1*PNL(I)
      RETURN
C
1      CCNTINUE
      DO 4 I=1,NTH
      AA1=C1(I)-CLB(I)

```

```

AA2=PNL(I)/AA1
AA3=C1(I)-CLB(I)
AA4=PNL(I)/AA3
F3=F3-AA2+AA4
5   GOTO (4,5)II
    AA2=AA2/AA1
    AA4=AA4/AA3
    F(I)=F(I)+AA2-AA4
C
C
100  GOTO (100,4)METH
    G(I,I)=G(I,I)+2.0*(AA4/AA3 - AA2/AA1)
C
C
4   CONTINUE

    RETURN
C
2   CONTINUE
    DATA CLB(1),CLB(2),CLB(3)/-999.0,-999.0,-999.0/
    DATA CUB(1),CUB(2),CUB(3)/999.0,999.0,999.0/
C
    DO 20 I=1,NTH
    21 IF(C1(I)-CLB(I))21,21,23
    25 IF(C1(I))25,26,27
        CLB(I)=100.*C1(I)
    26 GOTO 23
        CLB(I)=C1(I)-1.E10
    27 GOTO 23
        CLB(I)=0.
    23 IF(C1(I)-CUB(I))20,22,22
    22 IF(C1(I))28,29,24
    28 CUB(I)=0.
        GOTO 20
    29 CUB(I)=C1(I)+1.E10
        GOTO 20
    24 CUB(I)=100.*C1(I)
    20 CONTINUE
    DO 8 I=1,NTH
    8   PNL(I)=.0001*MIN(.C01+ABS(C1(I)),CUB(I)-CLB(I))
C
C
38  WRITE(NO,38)(I,CLB(I),CUB(I),PNL(I),I=1,NTH)
    FORMAT(///,2X,'PARAMETRO# COTA INF. COTA SUP.
* COEFICIENTE DE PENALIDAD :',
*/,(I9,2E16.6,E22.6),/)
C
    NCON=2*NTH
    RETURN
C
3   HY=0.
    DO 7 I=1,NTH

    7   HY=MIN(Y(I)/(C1(I)-CLB(I)),Y(I)/(C1(I)-CUB(I)),HY)
    CONTINUE
    IF (II-5)40,41,43
    40  H=MIN(1.,-.5/HY)
    RETURN
C
41  H=-1./HY
43  RETURN

```

END

```

-----!
SUBROUTINA PARA CALCULAR LAS MATRICES INVERSAS
-----!

```

```

C
C SUBROUTINE INVER(D,NOEQS,E)

```

```

C DIMENSION D(20,20),E(20,20)

```

```

C DO 2 I=1,NOEQS
C DO 2 J=1,NOEQS
C E(I,J)=0.0

```

```

C DO 6 M=1,NOEQS
C E(M,M)=1.0

```

```

C DO 13 MPIVRC=1,NOEQS
C NPIVCO=MPIVRC
C T=D(MPIVRC,NPIVCO)
C DO 1 N=1,NOEQS
C E(MPIVRC,N)=E(MPIVRC,N)/T
C D(MPIVRC,N)=D(MPIVRC,N)/T

```

```

C M=1
C CONTINUE
C IF(MPIVRC.EQ.M) GO TO 8
C CM=-D(M,NPIVCO)
C DO 11 N=1,NOEQS
C TM=D(MPIVRC,N)*CM
C TA=E(MPIVRC,N)*CM
C E(M,N)=E(M,N)+TA
C D(M,N)=D(M,N)+TM

```

```

C M=M+1
C IF(M.LE.NOEQS) GO TO 10

```

```

C 13 CONTINUE
C RETURN
C END

```

```

C
C

```

REGRESION NO-LINEAL POR EL METODO DE GAUSS-NEWTON O DE "LINEALIZACION"

OBSERVACIONES:		
OBS.	XI	YI
1	-0.500000E+01	0.127000E+03
2	-0.300000E+01	0.151000E+03
3	-0.100000E+01	0.379000E+03
4	0.100000E+01	0.421000E+03
5	0.300000E+01	0.460000E+03
6	0.500000E+01	0.426000E+03

PARAMETRO#	COTA INF.	COTA SUP.	COEFICIENTE DE PENALIDAD :
1	-0.999000E+03	0.999000E+03	0.500001E-01
2	-0.999000E+03	0.999000E+03	0.150001E-01
3	-0.999000E+03	0.999000E+03	0.201000E-04

PARAMETROS INICIALES:
 0.500000E+03 -0.150000E+03 -0.200000E+00

INCLUYENDO FUNCION DE PENALIDAD

ITERACION : 1

FUNCION: -0.1486950E+05 EVALUACION: 1
 VALOR DE LOS PARAMETROS:
 0.4999998E+03 -0.1500000E+03 -0.1999999E+00

FUNCION: -0.1339010E+05 EVALUACION: 2
 VALOR DE LOS PARAMETROS:
 0.5232332E+03 -0.1568720E+03 -0.1997128E+00

FUNCION: -0.1486815E+05 EVALUACION: 3
 VALOR DE LOS PARAMETROS:
 0.5464665E+03 -0.1637439E+03 -0.1994256E+00

ITERACION : 2

 FUNCION: -0.1339010E+05 EVALUACION: 4
 VALOR DE LOS PARAMETROS:
 0.5232328E+03 -0.1568720E+03 -0.1997127E+00

FUNCION: -0.1339010E+05 EVALUACION: 5
 VALOR DE LOS PARAMETROS:
 0.5234043E+03 -0.1570623E+03 -0.1995592E+00
 SIN FUNCION DE PENALIDAD

ITERACION : 3

 FUNCION: -0.1339009E+05 EVALUACION: 6
 VALOR DE LOS PARAMETROS:
 0.5232324E+03 -0.1568720E+03 -0.1997126E+00

FUNCION: -0.1339011E+05 EVALUACION: 7
 VALOR DE LOS PARAMETROS:
 0.5230321E+03 -0.1566320E+03 -0.1999549E+00

MAXIMO DE LA FUNCION OBJETIVO: -0.1339009E+05
 PARAMETROS:
 0.5232324E+03 -0.1568720E+03 -0.1997126E+00

EVALUACIONES DE LA FUNCION: 7
 EVALUACIONES DE DERIVADAS: 3

MODELO $Y = A_1 + A_2 \cdot \exp(A_3 \cdot X)$ ←

RESIDUALES (VALORES COMPUTADOS - VALORES OBSERVADOS)

-0.295774E+02 0.866394E+02 -0.473163E+02
 -0.262404E+02 -0.229350E+02 0.394395E+02

SUMA DEL CUADRADO DE LOS RESIDUALES: 0.1339009E+05

DESVIACION ESTANDAR: 0.6680842E+02

VALOR DE LOS PARAMETROS:

0.5232324E+03 -0.1568720E+03 -0.1997126E+00

VI.- INTERPRETACION GEOMETRICA DEL METODO DE GAUSS-NEWTON.

VI.- INTERPRETACION GEOMETRICA DEL METODO DE GAUSS-NEWTON.

Anteriormente observamos que la función suma de cuadrados $S(\underline{\theta})$ en un modelo lineal, es una función únicamente de los parámetros θ_i para $i=1,2, \dots, p$. En el espacio paramétrico p -dimensional, es decir, en el espacio geométrico generado por los parámetros θ_i la función $S(\underline{\theta})$ puede ser representada por los contornos de una superficie, semejantemente a curvas de nivel y si el modelo es lineal, las superficies de contorno son elípticas, concéntricas, y poseen un solo mínimo local y un solo mínimo global. [Ver figura (1).]

Por otra parte si el modelo no es lineal, entonces los contornos de las superficies no son elípticas, sino contrariamente, tienden a ser muy irregulares, en ciertos casos muy elongados, e incluso con elongaciones extendiéndose infinitamente, (Draper & Smith, han llamado a estas superficies, atendiendo a su forma, "banana-shaped"), en tales circunstancias la función $S(\underline{\theta})$ puede poseer más de un mínimo. [Ver figura (2).]

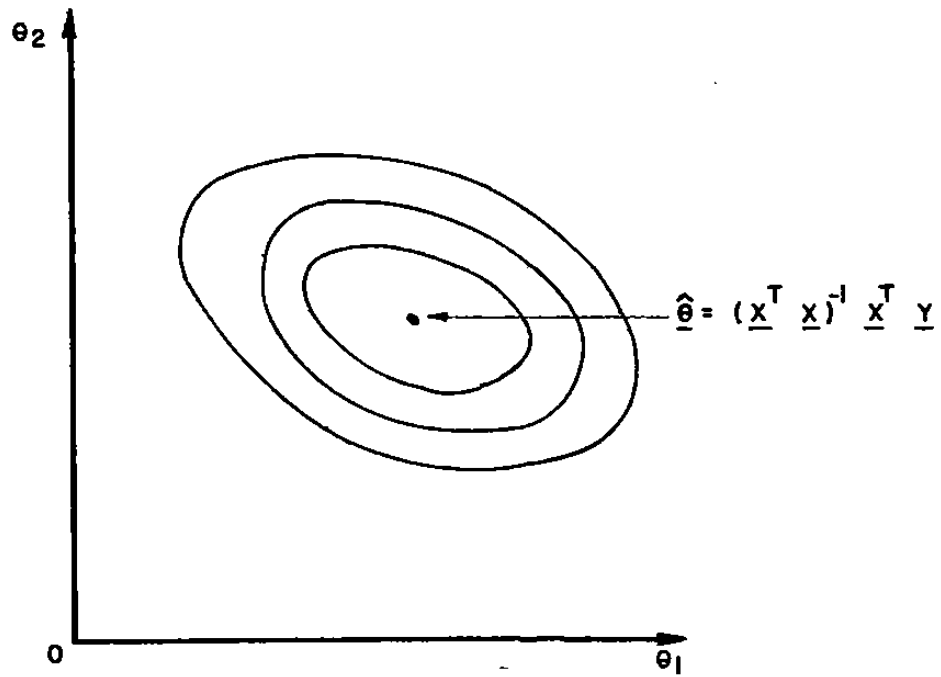


FIG. 1

SUPUESTOS CONTORNOS ELIPTICOS DE LAS
SUPERFICIES $S(\underline{\theta})$ EN UN MODELO LINEAL
CON DOS PARAMETROS.

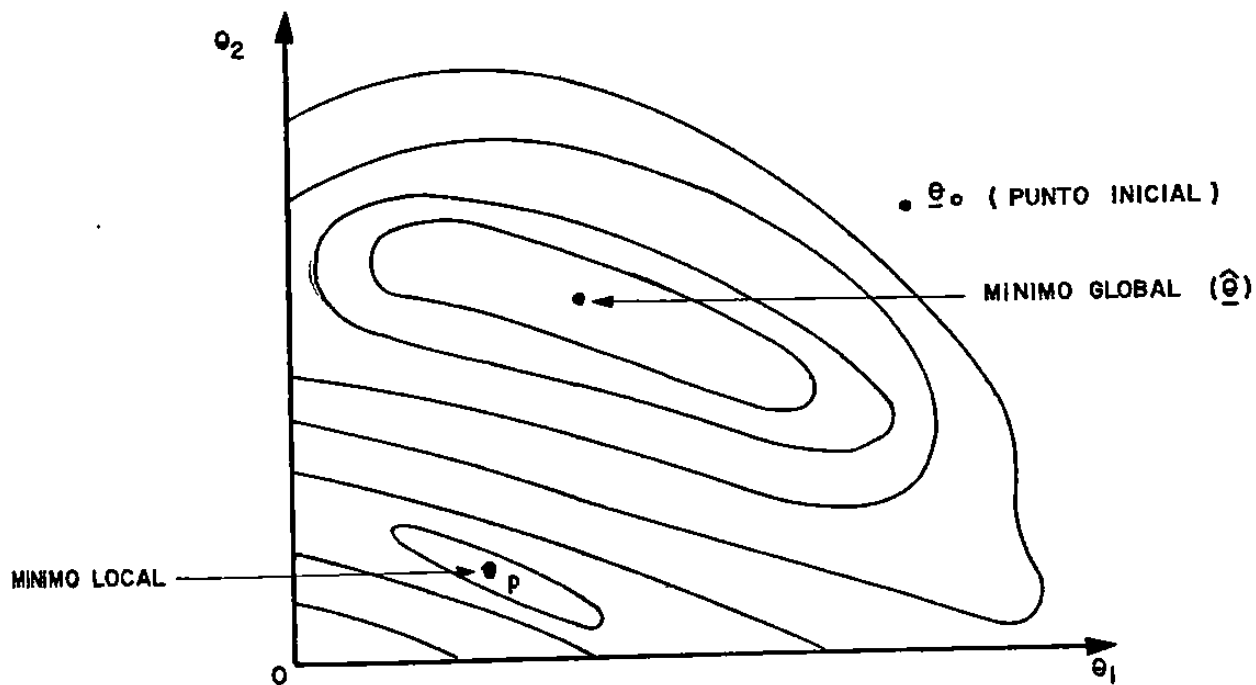


FIG. 2

CONTORNOS IRREGULARES DE LAS SUPERFICIES
DE $S(\underline{\theta})$ EN UN MODELO NO-LINEAL.
MOSTRANDO UN MINIMO LOCAL Y UNO GLOBAL

La forma precisa y la orientación de los contornos de la superficie $S(\theta)$ va a depender del modelo y de los datos. Cuando los contornos que rodean al estimador mínimo cuadrado $\hat{\theta}$ son muy elongados y varios posibles valores de θ son cercanamente "buenos" a $\hat{\theta}$ en el sentido de que los valores $S(\hat{\theta})$ son próximos a el valor \dots de $S(\hat{\theta})$, entonces se dice que el modelo esta mal condicionado, pudiendo indicar una sobreparametrización, es decir, que en el modelo se tienen más parámetros que los necesarios, tal situación puede también puede ser indicadora que el conjunto de datos es inadecuado para estimar los parámetros postulados en el modelo, tal mal acondicionamiento puede ocasionar dificultades computacionales durante el proceso de la estimación. El decidir cuando los datos o los parámetros son los causantes de dificultades, va a depender mucho del conocimiento práctico que se tenga del modelo que se este utilizando en alguna circunstancia particular.

Después del anterior vistazo a la geometría de la función suma de cuadrados $S(\theta)$, vamos ahora a ver el comportamiento geométrico del método de Gauss-Newton, en esencia sabemos que tal método convierte el problema de hallar el mínimo de $S(\theta)$ de un modelo no-lineal, en una serie de problemas que tratan el modelo original como si fuera lineal, de tal manera que el método, reemplaza un contorno irregular de la superficie $S(\theta)$ en un contorno elíptico.

El reemplazamiento que se realiza durante el proceso de aplicación del método de Gauss-Newton, puede realizarse bien o mal dependiendo de las siguientes tres situaciones:

- 1o. El modelo postulado.
- 2o. Los datos disponibles.

3o. El punto de inicio $\underline{\theta}_0$

El proceso se inicia en $\underline{\theta}_0$ hasta alcanzar un "óptimo", digamos $\underline{\theta}_1$; en esta primer etapa de la iteración, para alcanzar $\underline{\theta}_1$, se utiliza la técnica de mínimos cuadrados lineales, posteriormente se reemplaza a $\underline{\theta}_0$ por $\underline{\theta}_1$ y vuelve a repetirse la técnica de linealización y así sucesivamente hasta alcanzar la convergencia. [Ver figura (3).]

En teoría, la convergencia siempre es alcanzada, pero en la práctica se pueden encontrar casos en que el método diverge.

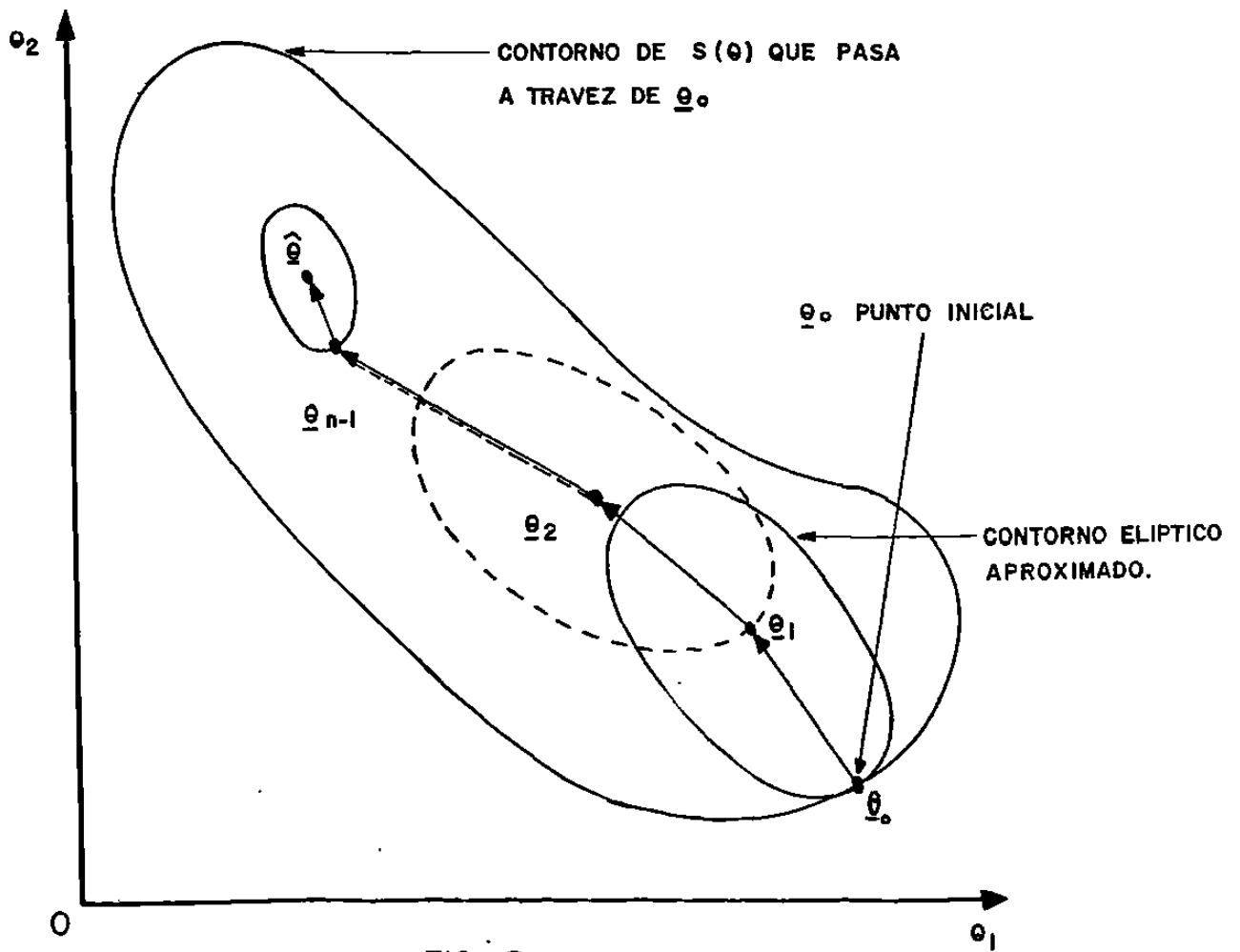


FIG. 3

METODO DE GAUSS-NEWTON EN UN SUPUESTO CASO BI-DIMENSIONAL .

VII.- METODO DEL COMPROMISO DE MARQUARDT.

VII.- METODO DEL COMPROMISO DE MARQUARDT.

Este método fue desarrollado por D. W. Marquardt y aparece publicado en:

Journal of Society for Industrial and Applied Mathematics, 2,
1963, pp 431-441.

Titulado como:

"An Algorithm for Least Square Estimation of Nonlinear Parameters"

En esencia este método representa un compromiso entre el método de Gauss-Newton y el de ascenso-descenso, conservando las mejores características de ambos y a la vez evitando sus más serias dificultades. Casi siempre converge, incluso para intentos iniciales relativamente malos, situación que no se presenta en el método de Gauss-Newton y cuando la convergencia es alcanzada, no lo hace tan lentamente como en el caso del

método de ascenso-descenso.

Antes de describir la idea central del método de Marquardt y tomando en cuenta que está relacionado con el método de ascenso-descenso, vamos a describir brevemente la idea central de este último método. Una descripción total, se encuentra en el trabajo:

Design Analysis of Industrial Experiments

Editado por:

D. L. Davies & Boyd, Edimburg Scotland, 1954.

El método se centra en la función suma de cuadrados

$$S(\underline{\theta}) = \sum_{u=1}^n [y_u - f(x_u; \underline{\theta})]^2$$

y utiliza un método iterativo para hallar el mínimo de la función; la idea básica es la de moverse a través del vector con componentes

$$-\frac{\partial S(\underline{\theta})}{\partial \theta_i} \quad ; \text{ para } i=1,2, \dots, p$$

desde un punto inicial $\underline{\theta}_0$. Es decir, la idea es de moverse a través de $-\nabla S(\underline{\theta})$ cuyos valores están continuamente cambiando. Se consigue esto, sin evaluar la totalidad de las derivadas, estimando los componentes del vector pendiente en varios lugares de la superficie $S(\underline{\theta})$ con funciones aproximadamente planas. El método se inicia en una región del espacio paramétrico y con la selección de una combinación de n niveles de los θ_i $i=1,2, \dots, p$ seleccionados convenientemente, por ejemplo, con una técnica de diseño de experimentos. Usando ahora los valores de $S(\underline{\theta})$ observados como una variable que depende de las combinaciones de $\theta_1, \theta_2, \dots, \theta_p$ y

a su vez considerando estas combinaciones como los valores correspondientes a los valores de las variables independientes, se obtiene el modelo:

$$\text{" } S(\underline{\theta}) \text{ observado " } = \beta_0 + \sum_{i=1}^p \beta_i \left(\frac{\theta_i - \bar{\theta}_i}{s_i} \right) + \varepsilon$$

donde $\bar{\theta}_i$ es la media de los niveles θ_{iu} ; $u=1,2,3, \dots, n$.

Tal modelo es resuelto por mínimos cuadrados lineales. Si observamos la anterior ecuación, se observa que corresponde a un hiperplano, significando que la superficie $S(\underline{\theta})$ es aproximada por un plano pero en la región paramétrica en la cual se hacen las corridas, es decir, en el sub-espacio de la selección de los n niveles de los θ_i ; $i=1,2,3, \dots, p$.

Entonces

$$\sum_{u=1}^n \left(\frac{\theta_{iu} - \bar{\theta}_i}{s_i} \right)^2 = k^2 ; \text{ para } k, \text{ una constante.}$$

así que los estimadores $-b_i$ de los $-\beta_i$ indican la dirección de descenso y los s_i son factores de escala. Los valores negativos de las estimaciones de los coeficientes indican una dirección de descenso (steepest descent).

Entre mejor sea la aproximación lineal, el máximo decremento en $S(\underline{\theta})$ se obtiene moviéndose a través de la línea que contiene los puntos tales que

$$\frac{\theta_i - \bar{\theta}_i}{s_i} \propto -b_i$$

Observamos que $-\nabla$ "S($\underline{\theta}$) observado" es el vector con componentes

$$\frac{\theta_i - \bar{\theta}_i}{s_i} ; i=1,2,3, \dots, p.$$

Considerando a λ como el factor de proporcionalidad, el vector del "escalón" de descenso que contiene los puntos $(\theta_1, \theta_2, \theta_3, \dots, \theta_p)$ tales que

$$\frac{\theta_i - \bar{\theta}_i}{s_i} = -\lambda b_i \quad \text{para } \lambda > 0$$

equivalentemente

$$\theta_i = \bar{\theta}_i - \lambda b_i s_i$$

un número de valores de λ son seleccionados y la trayectoria del descenso continúa y a su vez $S(\underline{\theta})$ sigue decreciendo. Cuando esto no es así, se selecciona otro conjunto de valores iniciales desde otro diseño de experimentos.

Volviendo ahora al método de Marquardt, como se ve anteriormente en el método de ascenso-descenso obtenemos una dirección vectorial, digamos δ_g que es obtenida mediante el gradiente. Si consideramos que este método utiliza una aproximación mediante un hiperplano que atenúa el contorno de $S(\underline{\theta})$ se puede considerar a δ_g como la mejor dirección "local" en la cual tenemos que movernos para alcanzar un menor valor de $S(\underline{\theta})$ pero no se le puede considerar a δ_g como la mejor dirección "global" para alcanzar el mínimo deseado.

El método de Gauss-Newton, deja otro vector de corrección, digamos δ Marquardt encontró que para un buen número de problemas prácticos, por él estudiados, el ángulo, digamos ϕ entre los vectores δ_g y δ se encuentra entre los valores $80^\circ < \phi < 90^\circ$ es decir que las dos direcciones casi siempre, están colocadas en ángulo recto!. De este modo el algoritmo de Marquardt proporciona, implícitamente, el método para la interpolación de los vectores δ_g y δ obtener, entonces una de las mejores direcciones que acortan el camino a la obtención del mínimo buscado.

Construimos ahora la siguiente ecuación matricial

$$\underline{A}^* \underline{b}^* = \underline{g}^*$$

donde

$$[A_{ij}^*] = [A_{ij}] / s_i s_j ; \text{ para } i \neq j$$

$$[A_{ii}^*] = 1 + \lambda$$

$$[g_i^*] = [g_i] / s_i$$

Arbitrariamente seleccionamos a $\lambda = 0.001$, aclarando que Marquardt propone un método para calcular este valor. Entonces

$$\begin{pmatrix} 1 + 0.001 & 0.008275 & -0.003757 \\ 0.008275 & 1 + 0.001 & -0.004161 \\ 0.003757 & -0.004161 & 1 + 0.001 \end{pmatrix} \begin{pmatrix} b_1^* \\ b_2^* \\ b_3^* \end{pmatrix} = \begin{pmatrix} 4.7527 \\ 1.7425 \\ -0.2785 \end{pmatrix}$$

resolviendo el sistema

$$\begin{pmatrix} b_1^* \\ b_2^* \\ b_3^* \end{pmatrix} = \begin{pmatrix} 0.9990831 & -0.0082437 & 0.0037155 \\ -0.0082437 & 0.9990863 & 0.0041221 \\ 0.0037155 & 0.0041221 & 0.9990320 \end{pmatrix} \begin{pmatrix} 4.7527 \\ 1.7425 \\ -0.2785 \end{pmatrix}$$

$$\begin{pmatrix} b_1^* \\ b_2^* \\ b_3^* \end{pmatrix} = \begin{pmatrix} 4.73300 \\ 1.70058 \\ -0.253389 \end{pmatrix}$$

reconsiderando la escala utilizada obtenemos

$$\underline{b}_1 = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 0.2539367 \\ 0.0349863 \\ -0.0011106 \end{pmatrix}$$

de tal manera que

$$\underline{\theta}_1 = \underline{\theta}_0 + \underline{b}_1$$

entonces

$$\underline{\theta}_1 = \begin{pmatrix} 500.25394 \\ -149.96501 \\ -0.2011106 \end{pmatrix}$$

Evaluando la función suma de cuadrados con los valores $\underline{\theta}_0, \underline{\theta}_1$ obtenemos

$$S(\underline{\theta}_0) = 14,869.486$$

$$S(\underline{\theta}_1) = 14,844.824$$

de tal manera que $S(\underline{\theta}_1) < S(\underline{\theta}_0)$.

Finalmente aplicamos alguna prueba de convergencia, de tal manera, que si esta es satisfecha entonces se reduce el valor de λ y se reemplaza $\underline{\theta}_0$ por $\underline{\theta}_1$ iniciando una nueva iteración y así sucesivamente hasta alcanzar la convergencia.

VIII.- DESCRIPCION DEL PROGRAMA PARA EL METODO DE MARQUARDT.

VIII.- DESCRIPCION DEL PROGRAMA PARA EL METODO DE MARQUARDT.

El programa esta basado en el trabajo original de

W. E. Ball
Washington University
St. Louis, Missouri. USA

Consta de un programa principal, dos subrutinas y una función.

- 1.- El programa principal llamado PRINCIPAL-BSOLVE, define unicamente un canal de salida para la impresion de resultados, los datos de entrada se encuentran implícitos en el programa.
- 2.- La subrutina BSOLVE es la principal, desarrolla los cálculos iniciales y coordina a todas las otras subrutinas.
- 3.- La subrutina FUNC, se utiliza para especificar el modelo, y el cuál es proporcionado por el usuario.
- 4.- La función ARCOS, genera una función (coseno) y es uti_

_lizada internamente en el programa.

Señalamos adicionalmente que el programa esta preparado para utilizar una subrutina, para el caso en que las derivadas se calculen analíticamente, actualmente en nuestro programa las derivadas se calculan numéricamente.

Descripción de algunos parámetros.

NN, es el número de puntos de entrada.

KK, número de incógnitas.

B, vector de incógnitas.

BMIN, vector de cotas superiores para los valores de B.

BMAX, vector de cotas inferiores para los valores de B.

X, es el vector de los valores de la variable independiente.

Y, vector de los valores de la variable dependiente.

PH, es la función objetivo de mínimos cuadrados.

X, son los valores calculados de la variable dependiente.

BV, es un vector código y en nuestro programa adopta el valor de 1, señal que es utilizada para indicar que las derivadas se calculan numéricamente y -1 para indicar el caso en que las derivadas se calculan analíticamente.

IX.- PROGRAMA PARA EL METODO DE MARQUARDT.

```

PROGRAM PRINCIPAL<BSOLVE
DIMENSION P(5C),A(10,10),AC(10,10),X(10)
DIMENSION B(10),Z(10),Y(10),BV(10),BMIN(10),BMAX(10)
COMMON X
C*****
C# LOS DATOS SE ENCUENTRAN IMPLICITOS, ES DECIR DENTRO DEL MISMO *
C# PROGRAMA; NO SE REQUIERE CANAL DE ENTRADA. *
C*****
      NO=6
      CPEN(UNIT=6,NAME='BSOLVE.CAT',STATUS='NEW')

C
C      LEER EL NUMERO DE DATOS
C
      NN=6
      KK=3

C
C      LECTURA DEL INTENTO INICIAL
C
      B(1)=400.00
      B(2)=-140.00
      B(3)=-0.1300

C
C      LECTURA DE LOS LIMITES DE LAS VARIABLES
C
      BMIN(1)=-1000.00
      BMIN(2)=-1000.00
      BMIN(3)=-1000.00
      BMAX(1)=1000.00
      BMAX(2)=1000.00
      BMAX(3)=1000.00

C
C      LECTURA DE LOS VALORES DE LA VARIABLE INDEP. Y DEPENDIENTE
C
      DATA X(1),X(2),X(3),X(4),X(5),X(6)/-5,-3,-1,1,3,5/
      DATA Y(1),Y(2),Y(3),Y(4),Y(5),Y(6)/127,151,379,421,460,426/

C
C
      FNU=0.0
      FLA=0.0
      TAU=0.0
      EPS=0.0
      PHMIN=0.0
      I=0
      KD=KK
      FV=0.0
      DO 100 J=1,KK
      BV(J)=1.0
100  CONTINUE
      ICON=KK
      ITER=0
      WRITE(NC,15)
15  FORMAT(10X,'ALGORITMO DE REGRESION BSOLVE')
C
200  CALL BSOLVE(KK,B,NN,Z,Y,PH,FNU,FLA,TAU,EPS,PHMIN,I,ICON,FV,DV,BV
      *BMIN,BMAX,P,FUNC,DERIV,KD,A,AC,GAMP)
C
      ITER=ITER+1
      WRITE(NO,1) ICON,PH,ITER
1  FORMAT(2X,78(' '),/,3X,'ICON=',I3,4X,'VALOR FUNCION MINI
      *MOS-CUAD. =',E15.8,4X,'# ITERACION =',I3)
      IF(ICON) 10,300,20

```

```

0      IF(ICON+1) 20,60,200
10     IF(ICON+2) 30,70,200
10     IF(ICON+3) 40,80,200
10     IF(ICON+4) 50,90,200
10     GC TO 95
10     WRITE(ND,4)
10     FORMAT(//,2X,'NO ES POSIBLE UN MEJORAMIENTO EN LA FUNCION')
10     GO TO 300
10     WRITE(ND,5)
10     FORMAT(//,2X,'MAS INCOGNITAS QUE FUNCIONES')
10     GC TO 300
10     WRITE(ND,6)
10     FORMAT(//,2X,'EL TOTAL DE VARIABLES SON CERO')
10     GC TO 300
90     WRITE(ND,7)
10     FORMAT(//,2X,'LAS CORRECCIONES SATISFACEN REQUERIMIENTOS DE
*CONVERGENCIA PERO EL FACTOR LAMBDA (FLA) ES GRANDE')
10     GO TO 300
15     WRITE(ND,8)
10     FORMAT(//,2X,'ESTO NO ES POSIBLE')
10     GC TO 300
100    WRITE(ND,2)
100    FORMAT(//,2X,75('-',),/,2X,'!',20X,'SOLUCIONES DE LA ECUACION',2EX,'!',
*,/,2X,75('-',))
100    DO 400 J=1,KN
100    WRITE(ND,3) J,B(J)
100    FORMAT(/,24X,'B(',I2,')=',E16.8)
300    CONTINUE
1000   STCP
1000   END
*****
*      SUBROUTINA QUE ES PROPORCIONADA POR EL USUARIO Y QUE CONTIENE
*      LA ECUACION DE REGRESION NO-LINEAL. EN ESTE PROGRAMA DE PRUE-
*      -BA, LA ECUACION ES:  Y= A + A * EXP( A * X )
*
*
*
*
*****
SUBROUTINE FUNCION (KK,B,NN,Z,FV)
COMMON X

DIMENSION X(25),Z(25),B(25)

DO 100 JJ=1,NN
Z(JJ)=B(1)+B(2)*(EXP(B(3)*X(JJ)))
100 CONTINUE

RETURN
END
*****
*      SUBROUTINA BSOLVE
*****
SUBROUTINE BSOLVE(KK,B,NN,Z,Y,PH,FNU,FLA,TAU,EPS,
*PHMIN,I,ICON,FV,DV,BV,BMIN,BMAX,P,FUNC,DERIV,KD,A,AC,GAMM)
DIMENSION B(10),Z(10),Y(10),BV(10),BMIN(10),BMAX(10)
DIMENSION P(10),A(10,10),AC(10,10),X(10),FV(10),OV(10)

K=KK
N=NN
KP1=K+1
KP2=KP1+1
KB11=K*N
KB12=KB11+K

```

```

KZI=KBI2+K
IF(FNU.LE.0.) FNU=10.0
IF(FLA.LE.0.) FLA=C.01
IF(TAU.LE.0.) TAU=C.001
IF(EPS.LE.0.) EPS=0.00002
IF(PHMIN.LE.0.) PHMIN=C.
120 KE=0
130 DO 160 I1=1,K
160 IF(BV(I1).NE.0.) KE=KE+1
IF(KE.GT.0) GO TO 170
162 ICON=-3
163 GO TO 2120
170 IF(N.GE.KE) GO TO 500
180 ICON=-2
190 GO TO 2120
500 I1=1
530 IF(I.GT.0) GO TO 1530
550 DO 560 J1=1,K
J2=KBI1+J1
P(J2)=B(J1)
J3=KBI2+J1
560 P(J3)=ABS(B(J1))+1.0E-02
GO TO 1030
590 IF(PHMIN.GT.PH.AND.1.GT.1) GO TO 625
DO 620 J1=1,K
N1=(J1-1)*N
C*****
C* SE PREPARA EL PROGRAMA PARA EN CASO DE QUE SE UTILICE DERIVADAS *
C* ANALITICAMENTE,ACTUALMENTE SE CALCULAN DERIVADAS NUMERICAMENTE *
C*****
IF(BV(J1)) 601,620,605
601 CALL DERIV(K,B,N,Z,P(N1+1),FV,DV,J1,JTEST)
IF(JTEST.NE.(-1)) GO TO 620
BV(J1)= 1.0
605 DO 606 J2=1,K
J3=KBI1+J2
606 P(J3)=B(J2)
J3=KBI1+J1
J4=KBI2+J1
DEN=0.001*MAX(P(J4),ABS(P(J3)))
IF (P(J3)+DEN.LE.BMAX(J1)) GO TO 55
P(J3)=P(J3)-DEN
DEN=-DEN
GO TO 56
55 P(J3)=P(J3)+DEN
56 CALL FUNCION(K,P(KBI1+1),N,P(N1+1),FV)
DO 610 J2=1,N
JB=J2+N1
610 P(JB)=(P(JB)-Z(J2))/DEN
620 CONTINUE
C
C COLOCAR ECUACIONES DE CORRECCION
C
625 DO 725 J1=1,K
N1=(J1-1)*N
A(J1,KP1)=0.
IF(BV(J1)) 630,692,630
630 DO 640 J2=1,N
N2=N1+J2
640 A(J1,KP1)=A(J1,KP1)+P(N2)*(Y(J2)-Z(J2))
650 DO 680 J2=1,K

```

```

660     A(J1,J2)=0.
665     N2=(J2-1)*N
670     DO 680 J3=1,N
672     N3=N1+J3
674     N4=N2+J3
680     A(J1,J2)=A(J1,J2)+P(N3)*P(N4)
        IF(A(J1,J1).GT.1.E-20) GO TO 725
692     DC 694 J2=1,KP1
694     A(J1,J2)=0.0
695     A(J1,J1)=1.0
725     CONTINUE
        GN=0.0
        DO 729 J1=1,K
729     GN=GN+A(J1,KP1)**2
C
C     ECUACIONES DE CORRECCION DE ESCALA
C
        DO 726 J1=1,K
726     A(J1,KP2)=SQRT(A(J1,J1))
        DO 727 J1=1,K
        A(J1,KP1)=A(J1,KP1)/A(J1,KP2)
        DO 727 J2=1,K
727     A(J1,J2)=A(J1,J2)/(A(J1,KP2)*A(J2,KP2))
730     FL=FLA/FNU
        GO TO 810
800     FL = FNU*FL
810     DO 840 J1=1,K
820     DC 830 J2=1,KP1
830     AC(J1,J2)=A(J1,J2)
840     AC(J1,J1)=AC(J1,J1)+FL
C
C     RESOLVER LAS ECUACIONES DE CORRECCION
C
        DO 930 L1=1,K
        L2=L1+1
        DO 910 L3=L2,KP1
910     AC(L1,L3)=AC(L1,L3)/AC(L1,L1)
        DO 930 L3=1,K
        IF(L1-L3) 920,930,920
920     DC 925 L4=L2,KP1
925     AC(L3,L4)=AC(L3,L4)-AC(L1,L4)*AC(L3,L1)
930     CONTINUE
C
        DN=0.0
        DG=0.0
        DO 1028 J1=1,K
        AC(J1,KP2)=AC(J1,KP1)/A(J1,KP2)
        J2=KB11+J1
        P(J2)=AMAX1(BMIN(J1),AMIN1(BMAX(J1),B(J1)+AC(J1,KP2)))
        DG=DG+AC(J1,KP2)*A(J1,KP1)*A(J1,KP2)
        DN=DN+AC(J1,KP2)*AC(J1,KP2)
1028     AC(J1,KP2)=P(J2)-B(J1)
        CCSG=DG/SQRT(DN*GN)
        JGAM=0
        IF(COSG) 1100,1110,1110
1100     JGAM=2
        CCSG=-COSG
1110     CONTINUE
        CCSG=MIN(CCSG,1.C)
        GAMM=ARCCOS(CCSG)*180.0/3.14159265
        IF(JGAM.GT.C) GAMM=18C.C-GAMM

```

```

1030 CALL FUNCION(K,P(KBI1+1),N,P(KZI+1),FV)
1500 PHI=0.0
      DO 1520 J1=1,N
      J2=KZI+J1
1520 PHI=PHI+(P(J2)-Y(J1))*#2
      IF(PHI.LT.1.E-10) GO TO 3000
      IF(I.GT.0) GC TO 1540
1521 ICON=K
      GO TO 2110
1540 IF(PHI.GE.PH) GO TO 1530
C
C PRUEBA DE EPSILON
C
1200 ICON=0
      DO 1220 J1=1,K
      J2=KBI1+J1
1220 IF(ABS(AC(J1,KP2))/(TAU+ABS(P(J2))).GT.EPS) ICON=ICON+1
      IF(ICON.EQ.0) GC TO 1400
C
C PRUEBA LAMBDA GAMMA
C
      IF(FL.GT.1.C.AND.GAMM.GT.90.0) ICON=-1
      GO TO 2105
C
C
C PRUEBA GAMMA EPSILON
1400 IF(FL.GT.1.C.AND.GAMM.LE.45.0) ICON=-4
      GO TO 2105
C
1530 IF(I1-2) 1531,1531,2310
1531 I1=I1+1
      GO TO (530,590,800) I1
2310 IF(FL.LT.1.0E+8) GC TO 800
1320 ICON=-1
C
2105 FLA=FL
      DO 2091 J2=1,K
      J3=KBI1+J2
2091 B(J2) = P(J3)
2110 DO 2050 J2=1,N
      J3=KZI+J2
      Z(J2)=P(J3)
2050 PH=PHI
      I=I+1
2120 RETURN
3000 ICON=0
      GC TO 2105
C
      END
C*****
C* SUBROUTINA FUNCION, PARA CALCULAR EL COSENO INVERSO. *
C*****
      FUNCTION ARCCS(Z)
      X=Z
      KEY=0
      IF(X.LT.(-1.)) X=-1
      IF(X.GT.1.) X=1
      IF(X.GE.(-1.).AND.X.LT.C.) KEY=1
      IF(X.LT.0.) X=ABS(X)
      IF(X.EQ.0.) GC TO 10

```

ARCOS=ATAN(SCRT(1.-X*X)/X)
IF(KEY.EQ.1) ARCOS=3.14159265-ARCOS
GO TO 999

10
C
999
ARCOS=1.5707963
RETURN
END

C
C

ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.75464789E+05	# ITERACION =	1
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.21340049E+05	# ITERACION =	2
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.16762031E+05	# ITERACION =	3
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13572878E+05	# ITERACION =	4
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13392149E+05	# ITERACION =	5
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13390444E+05	# ITERACION =	6
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13390161E+05	# ITERACION =	7
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13390111E+05	# ITERACION =	8
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13390108E+05	# ITERACION =	9
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13390095E+05	# ITERACION =	10
ICON=	3	VALOR FUNCION MINIMOS-CUAD. =	0.13390093E+05	# ITERACION =	11
ICON=	0	VALOR FUNCION MINIMOS-CUAD. =	0.13390092E+05	# ITERACION =	12

SOLUCIONES DE LA ECUACION

$$B(1) = 0.52329791E+03$$

$$B(2) = -0.15693806E+03$$

$$B(3) = -0.19967489E+00$$

X.- APENDICE A
METODO DE POWELL.

X.- APENDICE A.

METODO DE POWELL.

Este método fue desarrollado por M.J.D. Powell y el procedimiento esta descrito en el artículo:

Powell, M.J.D.

A Method for Minimizing a Sum of Squares of
Non-Linear Functions Without Calculating
Derivatives.

Computer Journal, 7, pp. 303-307, 1965.

Es la intención de este apéndice, hacer una descripción general del método. La idea central, es la modificación de las ecuaciones de Gauss-Newton (8), con el objetivo de evitar las dificultades que se involucran en la resolución del sistema de ecuaciones lineales y que se presentan en cada iteración del algoritmo.

El procedimiento también incorpora un método de inversión para matrices simétricas que cambian de iteración a iteración tan solo una fila y una columna, concretamente hacemos referencia a la matriz $Z_0(Z_0^T)$. Todas las derivadas son calculadas con la técnica de diferencias finitas. El algoritmo empieza con la selección de un punto inicial, digamos θ_0 y un conjunto de direcciones cuyos componentes vectoriales son

$$s_{ij} ; i=1,2, \dots, m ; j=1,2, \dots, m$$

estas direcciones. son paralelas a los ejes coordenados.

$$\underline{s}_{01} = (1, 0, 0, \dots, 0)$$

$$\underline{s}_{02} = (0, 1, 0, \dots, 0)$$

$$\underline{s}_{03} = (0, 0, 1, \dots, 0)$$

.

.

.

$$\underline{s}_{0m} = (0, 0, 0, \dots, 1)$$

donde m es el número de parámetros en el modelo

Evaluamos las ecuaciones de Gauss-Newton en el punto de inicio $\underline{\theta}_0$, donde las ecuaciones son

$$\underline{Z}_0^T \underline{Z}_0 \underline{b}_0 = \underline{Z}_0^T (\underline{y} - \underline{f}^0)$$

teniendo en cuenta que el procedimiento de Powell utiliza diferencias finitas para la evaluación de las derivadas se procede a resolver las ecuaciones para \underline{b}_0 . Ahora este vector se utiliza para calcular un nuevo vector de direcciones y se normaliza (norma unitaria) siendo las componentes

$$s_{k,i} = \frac{b_{k-1,i}}{\sqrt{b_{k-1,1}^2 + b_{k-1,2}^2 + \dots + b_{k-1,m}^2}}$$

para $i = 1, 2, 3, \dots, m$

y donde

$$\underline{b}_{k-1} = \begin{pmatrix} b_{k-1,1} \\ b_{k-1,2} \\ \vdots \\ b_{k-1,m} \end{pmatrix}$$

para k , que denota la k -ésima iteración.

m , denota el número de parámetros en el modelo.

Se inicia ahora una búsqueda uni-dimensional (problema de minimizar) para hallar un valor α_k en las nuevas direcciones $s_{k,i}$ y se utiliza la relación

$$b_{k,i} = b_{k-1,i} + \alpha_k s_{k,i} ; i=1,2, \dots, m$$

donde α_k va a representar la distancia recorrida en la k -ésima iteración en la dirección s_k .

Cuándo el mínimo uni-dimensional ha sido encontrado se realiza una prueba "global" de convergencia, si la prueba es satisfecha el procedimiento termina, si no, un componente del vector de direcciones, es reemplazado por uno nuevo, el reemplazado es aquel correspondiente al máximo de las siguientes cantidades

$$\left| \left[\underline{Z}_0^T (\underline{Y} - \underline{f}^0) \right]_i (b_{k,i}) \right| ; i=1,2, \dots, m$$

donde $\left[\underline{Z}_0^T (\underline{Y} - \underline{f}^0) \right]_i = i$ -ésimo elemento de $\underline{Z}_0^T (\underline{Y} - \underline{f}^0)$

$$b_{k,i} = i\text{-ésimo elemento de } \underline{b}_k$$

Se calculan ahora, las derivadas de \underline{Z}_0 en las nuevas direcciones, utilizando los valores hallados en la búsqueda uni-dimensional. Así las ecuaciones de Gauss-Newton, son actualizadas con las nuevas direcciones y se ha reemplazado además un elemento de $\underline{Z}(\underline{Y} - \underline{f}^0)$, además una fila y una columna de $\underline{Z}^T \underline{Z}$ resolviéndose

resolviéndose en seguida las ecuaciones y repitiendo el proceso hasta que finalmente una prueba de convergencia se satisfeca.

El procedimiento descrito anteriormente, es muy semejante a un método utilizado en programación matemática. Ver referencia (6). También desarrollado por Powell, tal método aparece publicado en el artículo

Powell M.J.D.

An Efficient Method for Finding the Minimum
of a Function of Several Variables Without
Calculating Derivatives.

The Computer Journal 7, pp. 155-162, 1964.

XI.- APENDICE B

DESCRIPCION DEL PROGRAMA PARA EL METODO DE POWELL.

XI.- APENDICE B .

DESCRIPCION DEL PROGRAMA PARA EL METODO DE POWELL.

El programa es basado en en el trabajo

E.R. Beals

Lawrence Radiation Laboratory

Berkeley, California. USA.

Este programa consiste de un programa principal y tres sub_rutinas.

1.- El programa principal controla el archivo de salida. Los datos de entrada se encuentran implícitos en el mismo programa. Se hace un único llamado a la subrutina principal.

2.- La subrutina principal es llamada SSQMIN, todas las salidas de resultados se hacen desde esta subrutina y además desde aquí se coordinan todos los cálculos.

3.- En la subrutina LINMIN es donde se desarrolla la búsqueda unidimensional, es decir se resuelve el problema de minimización y que se requiere en el método de Powell.

4.- La subrutina CALFUN especifica las diferencias entre los valores del modelo y los valores experimentales y debe de ser proporcionada por el usuario

Descripción de algunos parámetros:

M, es el número de puntos de entrada.

N, número de variables, minimamente dos.

F, es un vector de longitud m y que contiene los valores de

$$(\underline{y}_u - \underline{y}_u)$$

X, es un vector de incognitas (g's)

E, vector límite de exactitud absoluta y de longitud n, en los cambios de valor, en las variables en cada una de las iteraciones

ESCALE, parámetro delimitador del tamaño del paso.

IPRINT, es un controlador de impresiones para resultados intermedios.

MAXFUN, límite máximo de el número de evaluaciones de la función

W, es un vector de almacenamiento.

XII.- APENDICE C
PROGRAMA PARA EL METODO DE POWELL.

```

PROGRAM METCDD←DE←POWELL
C
C
C
PROGRAMA PRINCIPAL
C
DIMENSION F(6),X(3),E(3)
COMMON W(100)
C
NO=6
CPEN(UNIT=6,NAME='POWELL.DAT',STATUS='NEW')
C
DATA M,N,MAXFUN,IPRINT/6,3,100,1/
DATA ESCALE/1000/
DATA X(1),X(2),X(3)/500.0,-150.0,-0.2/
DATA E(1),E(2),E(3)/0.001,0.001,0.000001/
WRITE(NO,3)
3
FORMAT(15X,'ALGORITMO DE REGRESION DE POWELL')
C
CALL SSQMIN(M,N,F,X,E,ESCALE,IPRINT,MAXFUN,FF,NO)
C
WRITE(NO,4) FF
4
FORMAT(//,2X,'SUMA DE DIFERENCIAS AL CUADRADO=' ,1PE16.8)
WRITE(NO,5)
5
FORMAT(//,2X,78('*'),//,2X,'VALOR FINAL DE LOS
* COEFICIENTES:')

DC 200 J=1,N
200
WRITE(NO,6) J,X(J)
6
FORMAT(/,15X,'X(',12,')=' ,1PE16.8)
C
WRITE(NO,7)
7
FORMAT(2X,32('*'),'FIN DE TRABAJO',33('*'))
END

SUBROUTINE CALFUN(M,N,F,X)
C
DIMENSION F(M),X(N)
COMMON W(100)
C
F(1)=X(1)+X(2)*EXP(-5.*X(3))-127.0
F(2)=X(1)+X(2)*EXP(-3.*X(3))-151.0
F(3)=X(1)+X(2)*EXP(-1.*X(3))-379.0
F(4)=X(1)+X(2)*EXP( 1.*X(3))-421.0
F(5)=X(1)+X(2)*EXP( 3.*X(3))-460.0
F(6)=X(1)+X(2)*EXP( 5.*X(3))-426.0
C
RETURN
END

SUBROUTINE SSQMIN(M,N,F,X,E,ESCALE,IPRINT,MAXFUN,FF,NO)
C
DIMENSION F(M),X(N),E(N)
COMMON W(100)
LOGICAL STCP,MAXCAL,CONTIN,FIRST
C
WRITE(NO,12) M,N,MAXFUN,ESCALE

```

```

12  FORMAT(/,2X,'PARAMETRCOS',/,2X,'N=',I2,4X,'M=',I3,4X,
    *,'MAXFUN=',I5,4X,'ESCALE=',E10.4)
    WRITE(NO,13)
13  FORMAT(/,2X,'INTENTOS INICIALES')
    WRITE(NO,31) (I,X(I),I=1,N)
    WRITE(NO,18)
18  FORMAT(/,2X,'EXACTITUD DE LAS VARIABLES')
    WRITE(NO,23) (I,E(I),I=1,N)
23  FORMAT(/,3(2X,'E(',I2,')=' ,E16.8))
C
C  INCIALIZA
C
    STCP=.FALSE.
    MAXCAL=.FALSE.
    IPP=IPRINT*(IPRINT-1)
    ITC=0
    IPC=0
    MPLUSN=M+N
    KST=N+MPLUSN
    NPLUS=N+1
    KINV=NPLUS*(MPLUSN+1)
    KSTORE=KINV-MPLUSN-1
    NN=N+N
    K=NN
C
C  EVALUACION INICIAL DE LA FUNCION
C
    CALL CALFUN(M,N,F,X)
C
    MC=1
    DUMM1=0.0
    FF=0.0
    DO 1 I=1,M
    K=K+1
    W(K)=F(I)
    FF=FF+F(I)*F(I)
1   CCNTINUE
    FOLD=FF
100  FIRST=.TRUE.
    K=KST
    I=1
C
C  CALCULO DE LOS COMPONENTES DEL GRADIENTE EN LAS DIRECCIONES
C  DE LAS COORDENADAS.
C
2   XDUMMY=X(I)
    ISMALL=0
    DUMMY=ABS(X(I)*1.E-6)+E(I)
5   X(I)=X(I)+DUMMY
    CALL CALFUN(M,N,F,X)
    IF(DUMM1.EQ.0) MC=MC+1
    X(I)=XDUMMY
    DO 3 J=1,N
    K=K+1
    W(K)=0.
    W(J)=0.
3   CONTINUE
    SUM=0.
    KK=NN
    DO 4 J=1,M
    KK=KK+1

```

```

C
C      FPLUS-FBEST
C
C      F(J)=F(J)-W(KK)
C      SUM=SUM+F(J)*F(J)
4     CONTINUE
C      IF(SUM.GT.FF*1.E-12) GOTO 6
C
C      WRITE(NO,7) I
7     FORMAT(5X,'EL',I3,'-ESIMO COMPONENTE DEL PASO INICIAL ES TAMBIEN
* DOBLEMENTE PEQUENIO')
C      DUMMY=DUMMY*2.0
C
C      ISMALL=ISMALL+1
C      K=K-N
C      IF(ISMALL.LT.15) GOTO 5
C      ITC=0
C      K=NN
C      DO 8 I=1,M
C      K=K+1
C      F(I)=W(K)
8     CONTINUE
C      GO TO 10
C
C LA SUMA ES UTILIZADA PARA NORMALIZAR G(I,K) & D(I,J)
C
6     SUM=1.0/SQRT(SUM)
C      ISMALL=0
C      J=K-N+1
C
C W(J) ES D(I,I) . NOTAR QUE D(I,J)=0.0 NO ES IGUAL A I
C
C      W(J)=DUMMY*SUM
C      DO 9 J=1,M
C      K=K+1
C
C W(J) ES G(I,K) EN LAS DIRECCIONES DE LAS COORDENADAS
C
C      W(K)=F(J)*SUM
C      KK=NN+J
C      DO 11 II=1,I
C      KK=KK+MPLUSN
C
C      W(II) ES G*GT(I,II)
C
C      W(II)=W(II)+W(KK)*W(K)
11    CONTINUE
9     CONTINUE
C      ILESS=I-1
C      IGAMAX=N+I-1
C      INCINV=N-ILESS
C      INCINP=INCINV+1
C      IF(ILESS.GT.0) GO TO 14
C
C INVERSA DE G*GT(II,JJ) II,JJ=1,I .POR EL METODO DE HOUSEHOLDER
C LLAMAR AL BLOQUE SUPERIOR (I-1)X(I-1) ,QUE ESTA LISTO
C
C      W(KINV)=1.0
C      GOTO 15
14    IF(ISAME.EQ.3.AND.I.NE.3) MC=MC+1
C      B=1.0

```

```

DO 16 J=NPLUS,IGAMAX
W(J)=0.
16 CONTINUE
KK=KINV
DO 17 II=1,ILESS
IIP=II+N
C
C W(IIP)=W(N+II) ES LA SUMA DE G-1(II,J)*G*GT(J,I) J=1,N
C
W(IIP)=W(IIP)+W(KK)*W(II)
JL=II+1
IF(JL.GT.ILESS) GOTO 19
DO 20 JJ=JL,ILESS
KK=KK+1
JJP=JJ+N
W(IIP)=W(IIP)+W(KK)*W(JJ)
W(JJP)=W(JJP)+W(KK)*W(II)
20 CONTINUE
C
C B ES G*G(I,I)-SUM DE G*GT(I,II)*G-1(II,JJ)*G*GT(JJ,I)
C LA CUAL ES A0
C
19 B=B-W(II)*W(IIP)
KK=KK+INCINP
17 CONTINUE
B=1.0/B
KK=KINV
DO 21 II=NPLUS,IGAMAX
BB=-B*W(II)
DO 22 JJ=II,IGAMAX
C
C W(KK) ES G-1(II,JJ) LA CUAL ES IGUAL A A1-1+A1-1*A2*A0-1*A3*A1-1
C
W(KK)=W(KK)-BB*W(JJ)
KK=KK+1
22 CONTINUE
C W(KK) ES G-1(I,II) LA CUAL ES IGUAL -A0-1*A1
C LA CUAL ES IGUAL G-1(II,I)
W(KK)=BB
KK=KK+INCINP
21 CONTINUE
C
C W(KK) ES G-1(I,I) LA CUAL ES IGUAL A A0-1
C
W(KK)=B
15 IF(.NOT.FIRST) GOTO 27
I=I+1
IF(I.LE.N) GOTO 2
C
C EMPEZAMOS CON LA ITERACION C-ESIMA.
C
FIRST=.FALSE.
ISAME=0
FF=0.
KL=NN
DO 26 I=1,M
KL=KL+1
F(I)=W(KL)
FF=FF+F(I)*F(I)
26 CONTINUE
CONTIN=.TRUE.

```

```

27     IPC=IPC-IPRINT
      IF(IPC.GE.0) GOTC 29
      IF(DUMM1.EQ.1.0) GCTO 29
C
C  IMPRESION DE CADA UNA DE LAS ITERACIONES.
C
28     WRITE(NO,30) ITC,MC,FF
30     FORMAT(//,2X,78('*'),/,2X,'ITERACION NO.',I3,4X,
      *'NUMERO DE EVALUACIONES DE LA FUNCION =',I4,//,2X,
      *'SUMA DE CUADRADOS =',E16.8,//,2X,'COEFICIENTES')
      WRITE(NO,31) (I,X(I),I=1,N)
31     FORMAT(/,3(2X,'X(',I2,')=' ,E16.8))
      WRITE(NO,32)
32     FORMAT(/,2X,'VALORES DE LA FUNCION:')
      WRITE(NO,35) (J,F(J),J=1,M)
35     FORMAT(/,3(2X,'F(',I2,')=' ,E16.8))
      IPC=IPP
      IF(STOP) GO TO 33
C
C
C  PRUEBAS DE CONVERGENCIA
C  EL MAXIMO DE STEP(I)/E(I) ES MENOR O IGUAL QUE 1.0
C  ( CONTIN ES FALSO )
C  EL MAXIMO DEL 1-ESIMO COMPONENTE, EN EL PASO ACTUAL
C  ES TOMADO/ E(I)
C
C
      CHANGE=0.0
29     IF(CHANGE.NE.0.0) ISAME=0
      ISAME=ISAME+1
      IF(ISAME.LT.N) GO TO 291
      IF(IPRINT.LE.0) GO TO 33
      IF(FF.GE.FCLD) GO TO 10
      FOLD=FF
      K=NN
      DO 293 I=1,M
      K=K+1
      W(K)=F(I)
293    CONTINUE
      DUMM1=1.0
      GO TO 100
291    IF(CONTIN) GC TO 34
      IF(CHANGE.GT.1.0) GO TO 36
10     IF(IPRINT.LE.0) GC TO 33
C
C  SALIDA DE LAS IMPRESIONES.
C
      WRITE(NO,38)
38     FORMAT(//,10X,'VALORES FINALES SSQMIN DE FUNCIONES Y
      * VARIABLES')
      STOP=.TRUE.
      GC TO 28
33     RETURN
C
36     CCNTIN=.TRUE.
C
C  SE INICIA LA PROXIMA ITERACION.
C
34     ITC=ITC+1
      K=N
      KK=KST

```

```

C
C CALCULO DE P
C
      DO 39 I=1,N
      K=K+1
      W(K)=0.0
      KK=KK+N
      W(I)=0.0
      DO 40 J=1,M
      KK=KK+1

C
C W(I) ES LA SUMA DE G(I,K)*F(K) LA CUAL ES -P(I)
C
      W(I)=W(I)+W(KK)*F(J)
40      CCONTINUE
39      CONTINUE
      DM=0.
      K=KINV

C
C CALCULO DE Q
C
      DO 41 II=1,N
      IIP=II+N

C
C
C
      W(IIP)=W(IIP)+W(K)*W(II)
      JL=II+1
      IF(JL.GT.N) GO TO 43
      DO 44 JJ=JL,N
      JJP=JJ+N
      K=K+1
      W(IIP)=W(IIP)+W(K)*W(JJ)
      W(JJP)=W(JJP)+W(K)*W(II)
44      CCONTINUE
      K=K+1

C
C EL MAXIMO DE P(I)*Q(I) EL INDICE KL DE LA DIRECCION
C DE C(I,J) ES REMPLAZADO POR STEP(J)
C
43      IF(DM.GE.ABS(W(II)*W(IIP))) GO TO 41
      DM = ABS (W(II)*W(IIP))
      KL=II
41      CONTINUE
      II=N+MPLUSN*KL
      CHANGE=0.
      DO 46 I=1,N
      JL=N+I
      W(I)=0.0
      DO 47 J=NPLUS,NA
      JL=JL+MPLUSN

C
      W(I) ES LA SUMA DE (-C(J)*D(J,I) J=1,N EL CUAL ES -STEP(I)
C
      W(I)=W(I)+W(J)*W(JL)
47      CONTINUE
      II=II+1

C
C INTERCAMBIANDO KL Y N FILAS DE D(I,J) COLOCAR XBEST EN D(N,J)
C
      W(II)=W(JL)

```

```

      W(JL)=X(I)
C
C CHANGE ES EL MAXIMO DE ABS(STEP(I)/E(I))
C
      IF(ABS(E(I)*CHANGE).GT.ABS(W(I))) GO TO 46
      CHANGE=ABS(W(I)/E(I))
46    CONTINUE
      DO 49 I=1,M
      II=II+1
      JL=JL+1
C
C INTERCAMBIANDO KL Y N FILAS DE G COLOCAMOS FBEST EN G(N,K)
C
      W(II)=W(JL)
      W(JL)=F(I)
49    CONTINUE
      FC=FF
      ACC=0.1/CHANGE
      IT=3
      XC=0.0
      XL=0.0
      IS=3
      XSTEP=-MIN(0.5,ESCALE/CHANGE)
      IF(CHANGE.LE.1.0) CONTIN=.FALSE.
C
C BUSQUEDA LINEAL
C
51    CALL LINMIN(IT,XC,FC,6,ACC,0.1,XSTEP)
      IF(IT.NE.1) GO TO 53
      MC=MC+1
      IF(MC.LE.MAXFUN) GO TO 54
      WRITE( NO,56) MAXFUN
56    FORMAT(5X,I6,'LLAMADO DE CALFUN')
      MAXCAL=.TRUE.
      GO TO 53
54    XL=XC-XL
      DO 57 J=1,N
      X(J)=X(J)+XL*W(J)
57    CONTINUE
      XL=XC
      CALL CALFUN(M,N,F,X)
      FC=0.0
      DO 58 J=1,M
      FC=FC+F(J)*F(J)
58    CONTINUE
      IF(IS.NE.3) GO TO 59
      K=N
C
C DETERMINACION DEL SEGUNDO MEJOR PUNTO
C
      IF(FC-FF) 61,51,62
61    IS=2
      FMIN=FC
      FSEC=FF
      GO TO 63
62    IS=1
      FMIN=FF
      FSEC=FC
      GO TO 63
59    IF(FC.GE.FSEC) GO TO 51
      K=KSTORE

```



```

        IF (IS.EQ.2) GO TO 74
        K=N
74      IF (FC-FMIN) 65,51,66
66      FSEC=FC
        GO TO 63
65      IS=3-IS
        FSEC=FMIN
        FMIN=FC
63      DO 67 J=1,N
        K=K+1
        W(K)=X(J)
67      CONTINUE
        DO 68 J=1,M
        K=K+1
        W(K)=F(J)
68      CONTINUE
        GO TO 51
53      K=KSTORE
        KK=N
C
C SI IS=2  XBEST Y FBEST ESTAN EN W(N+ ) LA SEGUNDA MEJOR X Y Y ESTAN
C W(KSTORE+ )=D(N,J) Y G(N,K)
C SI IS NO ES 2 XBEST Y FBEST ESTAN EN W(KSTORE+ ) Y LA SEGUNDA MEJOR
C ESTA EN W(N+ )
C
        IF (IS.NE. 2) GO TO 69
        K=N
        KK=KSTORE
69      SUM=0.
        DM=0.0
        JJ=KSTORE
        DO 71 J=1,N
        K=K+1
        KK=KK+1
        JJ=JJ+1
C
C      XBEST ESTA EN X
C      XBEST-XSECCND EN D(N,J)
C
        X(J)=W(K)
        W(JJ)=W(K)-W(KK)
71      CONTINUE
        DO 72 J=1,M
        K=K+1
        KK=KK+1
        JJ=JJ+1
C
C FBEST ESTA EN F
C FBEST-FSECCND EN G(N,K)
C
        F(J)=W(K)
        W(JJ)=W(K)-W(KK)
        SUM=SUM+W(JJ)*W(JJ)
        DM=DM+F(J)*W(JJ)
72      CONTINUE
        IF (MAXCAL) GO TO 10
        J=KINV
        KK=NPLUS-KL
        DO 76 I=1,KL
        K=J+KL-I
        J=K+KK

```

```

C
C      INTERCAMBIANDO KL Y N FILAS DE G-1
C
      W(I)=W(K)
      W(K)=W(J-1)
76    CONTINUE
      IF(KL.GE.N) GO TO 78
      KL=KL+1
      JJ=K
      DO 79 I=KL,N
      K=K+1
      J=J+NPLUS-I
      W(I)=W(K)
      W(K)=W(J-1)
79    CONTINUE
      W(JJ)=W(K)
      B=1.0/W(KL-1)
      W(KL-1)=W(N)
      GO TO 88
78    B=1.0/W(N)
88    K=KINV
C
C      DETERMINAR A1-1 DESDE G-1 PARA USARLA EN EL CALCULO DE LA NUEVA G-1
C
      DO 80 I=1,ILESS
      BB=B*W(I)
      DO 81 J=I,ILESS
C
C      W(K) ES G-1(I,J) LA CUAL ES A1-1=B1-B2*B4-1*B3
C
      W(K) = W(K) - BB*W(J)
      K=K+1
81    CONTINUE
      K=K+1
80    CCNTINUE
      IF(FMIN.LT.FF) GO TO 82
      CHANGE=0.0
      GO TO 84
82    FF=FMIN
C
C      CHANGE ES EL MAXIMO DE LOS COMPONENTES DEL ACTUAL PASO
C      DIVIDIDO ENTRE LOS COMPONENTES DE E
C      LA SUMA ES USADA PARA NORMALIZAR G(N,K) Y D(N,J)
C
      CHANGE=ABS(XC)*CHANGE
84    XL=-DM/FMIN
      SUM=1.0/SQRT(SUM+DM*XL)
      K=KSTORE
      DO 85 I=1,N
      K=K+1
C
C      W(K) ES D(N,J) ESTE PASO PROPIAMENTE NORMALIZADO
C
      W(K)=SUM*W(K)
      W(I)=0.0
85    CONTINUE
      DO 86 I=1,M
      K=K+1
C
C      W(K) ES G(N,K) LA CUAL ES (FBEST-FSECOND+(SUMA DE (FBEST-FSECOND)*FBEST/
C      FMIN)*FBEST) NORMALIZADO.

```

```

C      W(K)=SUM*(W(K)+XL*F(I))
      KK=NN+I
      DO 87 J=1,N
      KK=KK+MPLUSA
C
C      W(J) ES LA N-ESIMA FILA DE G*GT
C
      W(J)=W(J)+W(KK)*W(K)
87     CONTINUE
86     CONTINUE
      DUMM1=0.0
      GO TO 14
C
      END
C
C
C
C      SUBROUTINE LINMIN(ITEST,X,F,MAXFUN,ABSACC,RELACC,XSTEP)
C
      GO TO (1,2,2) ,ITEST
C
2     IS=6-ITEST
      ITEST=1
      IINC=1
      XINC=XSTEP+XSTEP
      MC=IS-3
      IF(MC) 4,4,15
3     MC=MC+1
      IF(MAXFUN. GE. MC) GO TO 15
      ITEST=4
43    X=DB
      F=FB
      IF(FB.LE.FC) GO TO 15
      X=DC
      F=FC
15    RETURN
C
1     GO TO (5,6,7,8),IS
8     IS=3
4     DC=X
      FC=F
      X=X+XSTEP
      GO TO 3
7     IF(FC-F) 9,10,11
10    X=X+XINC
      XINC=XINC+XINC
      GO TO 3
9     DB=X
      FB=F
      XINC=-XINC
      GO TO 13
11    DB=DC
      FB=FC
      DC=X
      FC=F
13    X=DC+DC-DB
      IS=2
      GO TO 3
6     DA=DB
      DB=DC

```

```

FA=FB
FB=FC
32 DC=X
FC=F
GO TO 14
5 IF(FB.LT.FC) GO TO 16
IF(F.GE.FB) GO TO 32
FA=FB
19 DA=DB
FB=F
DB=X
GO TO 14
16 IF(FA.LE.FC) GO TO 21
XINC=FA
FA=FC
FC=XINC
XINC=DA
DA=DC
DC=XINC
21 XINC=DC
IF((D-DB)*(D-DC).LT.0.0) GO TO 32
IF(F.GE.FA) GO TO 24
FC=FB
DC=DB
GO TO 19
24 FA=F
DA=X
14 IF(FB.GT.FC) GO TO 29
IINC=2
XINC=DC
IF(FB.EQ.FC) GO TO 45
29 D=(FA-FB)/(DA-DB)-(FA-FC)/(DA-DC)
IF(D*(DB-DC).LT.0.0) GO TO 33
D=0.5*(DB+DC-(FB-FC)/D)
IF((ABS(D-X).GT.ABS(ABSACC)).AND.(ABS(D-X).GT.ABS(D*RELACC)))
* GO TO 36
ITEST=2
GO TO 43
36 IS=1
X=D
IF((DA-DC)*(DC-D)) 3,26,38
38 IS=2
GO TO (39,40),IINC
33 IS=2
GO TO (41,42),IINC
39 IF(ABS(XINC).GE.ABS(DC-D)) GO TO 3
41 X=DC
GO TO 10
40 IF(ABS(XINC-X).GT.ABS(X-DC)) GO TO 3
42 X=0.5*(XINC+DC)
IF((XINC-X)*(X-DC).GT.0.0) GO TO 3
GO TO 26
45 X=0.5*(DB+DC)
IF((DB-X)*(X-DC).GT.0.0) GO TO 3
26 ITEST=3
GO TO 43
C
END

```

PARAMETROS

N= 6 M= 3 MAXFUN= 100 \ ESCALE=0.1000E+04

INTENTOS INICIALES

X(1)= 0.50000000E+03 X(2)= -0.15000000E+03 X(3)= -0.20000000E+00

EXACTITUD DE LAS VARIABLES

E(1)= 0.10000000E-02 E(2)= 0.10000000E-02 E(3)= 0.10000000E-05

ITERACION NO. 0 NUMERO DE EVALUACIONES DE LA FUNCION = 4

SUMA DE CUADRADOS = 0.14869480E+05

COEFICIENTES

X(1)= 0.50000000E+03 X(2)= -0.15000000E+03 X(3)= -0.20000000E+00

VALORES DE LA FUNCION:

F(1)= -0.34742249E+02 F(2)= 0.75682159E+02 F(3)= -0.62210419E+02
F(4)= -0.43809601E+02 F(5)= -0.42321747E+02 F(6)= 0.18818085E+02

ITERACION NO. 1 NUMERO DE EVALUACIONES DE LA FUNCION = 10

SUMA DE CUADRADOS = 0.13394751E+05

COEFICIENTES

X(1)= 0.52104175E+03 X(2)= -0.15426363E+03 X(3)= -0.20293368E+00

VALORES DE LA FUNCION:

F(1)= -0.31486572E+02 F(2)= 0.86470306E+02 F(3)= -0.46929840E+02
F(4)= -0.25888641E+02 F(5)= -0.22878082E+02 F(6)= 0.39117706E+02

ITERACION NO. 2 NUMERO DE EVALUACIONES DE LA FUNCION = 14

SUMA DE CUADRADOS = 0.13392745E+05

COEFICIENTES

X(1)= 0.52397223E+03 X(2)= -0.15767288E+03 X(3)= -0.19969867E+00

VALORES DE LA FUNCION:

F(1)= -0.30981842E+02 F(2)= 0.85923105E+02 F(3)= -0.47551819E+02
F(4)= -0.26158295E+02 F(5)= -0.22638733E+02 F(6)= 0.39880157E+02

ITERACION NO. 3 NUMERC DE EVALUACIONES DE LA FUNCION = 17

SUMA DE CUADRADOS = 0.13390883E+05

COEFICIENTES

X(1)= 0.52226117E+03 X(2)= -0.15567288E+03 X(3)= -0.20114706E+00

VALORES DE LA FUNCION:

F(1)= -0.30335541E+02 F(2)= 0.86628906E+02 F(3)= -0.47096344E+02
F(4)= -0.26046906E+02 F(5)= -0.22880432E+02 F(6)= 0.39319824E+02

ITERACION NO. 4 NUMERC DE EVALUACIONES DE LA FUNCION = 20

SUMA DE CUADRADOS = 0.13390156E+05

COEFICIENTES

X(1)= 0.52366827E+03 X(2)= -0.15730629E+03 X(3)= -0.19943151E+00

VALORES DE LA FUNCION:

F(1)= -0.29720856E+02 F(2)= 0.86525940E+02 F(3)= -0.47356873E+02
F(4)= -0.26196442E+02 F(5)= -0.22810608E+02 F(6)= 0.39633789E+02

ITERACION NO. 5 NUMERC DE EVALUACIONES DE LA FUNCION = 24

SUMA DE CUADRADOS = 0.13390131E+05

COEFICIENTES

X(1)= 0.52336621E+03 X(2)= -0.15695694E+03 X(3)= -0.19975123E+00

VALORES DE LA FUNCION:

F(1)= -0.29756622E+02 F(2)= 0.86585358E+02 F(3)= -0.47293732E+02
F(4)= -0.26171234E+02 F(5)= -0.22837891E+02 F(6)= 0.39552131E+02

ITERACION NO. 6 NUMERC DE EVALUACIONES DE LA FUNCION = 28

SUMA DE CUADRADOS = 0.13390091E+05

COEFICIENTES

X(1)= 0.52332867E+03 X(2)= -0.15697867E+03 X(3)= -0.19964160E+00

VALORES DE LA FUNCION:

F(1)= -0.29619598E+02 F(2)= 0.86602264E+02 F(3)= -0.47336792E+02
F(4)= -0.26240662E+02 F(5)= -0.22915710E+02 F(6)= 0.39475861E+02

ITERACION NO. 7 NUMERO DE EVALUACIONES DE LA FUNCION = 35

78

SUMA DE CUADRADOS = 0.13390091E+05

COEFICIENTES

X(1)= 0.52332867E+03 X(2)= -0.15697867E+03 X(3)= -0.19964160E+00

VALORES DE LA FUNCION:

F(1)= -0.29619598E+02 F(2)= 0.86602264E+02 F(3)= -0.47336792E+02
F(4)= -0.26240662E+02 F(5)= -0.22915710E+02 F(6)= 0.39475861E+02

ITERACION NO. 8 NUMERO DE EVALUACIONES DE LA FUNCION = 41

SUMA DE CUADRADOS = 0.13390091E+05

COEFICIENTES

X(1)= 0.52332867E+03 X(2)= -0.15697867E+03 X(3)= -0.19964160E+00

VALORES DE LA FUNCION:

F(1)= -0.29619598E+02 F(2)= 0.86602264E+02 F(3)= -0.47336792E+02
F(4)= -0.26240662E+02 F(5)= -0.22915710E+02 F(6)= 0.39475861E+02

VALORES FINALES SSCMIN DE FUNCIONES Y VARIABLES

ITERACION NO. 8 NUMERO DE EVALUACIONES DE LA FUNCION = 41

SUMA DE CUADRADOS = 0.13390091E+05

COEFICIENTES

X(1)= 0.52332867E+03 X(2)= -0.15697867E+03 X(3)= -0.19964160E+00

VALORES DE LA FUNCION:

F(1)= -0.29619598E+02 F(2)= 0.86602264E+02 F(3)= -0.47336792E+02
F(4)= -0.26240662E+02 F(5)= -0.22915710E+02 F(6)= 0.39475861E+02

SUMA DE DIFERENCIAS AL CUADRADO= 1.33900908E+04

VALOR FINAL DE LOS COEFICIENTES:

X(1)= 5.23328674E+02

X(2)= -1.56978668E+02

X(3)= -1.99641600E-01

*****FIN DE TRABAJO*****

CONCLUSIONES

CONCLUSIONES

Tiene el propósito esta última sección, de presentar a manera de conclusiones, las opiniones del autor de la tesis acerca de los temas que se han expuesto, aclaramos inicialmente que se quizá conservar un enfoque práctico en cada uno de los métodos de regresión no lineal y que en el transcurso del trabajo se presentarón, más que un enfoque teórico, no significando con esto, que se soslaye tan importante aspecto, aclaramos además, que fue también el enfoque práctico una de las razones principales tomadas en cuenta, para incluir la presentación de las rutinas computacionales.

Por otro lado, es nuestra opinión, que en el caso de los métodos de la regresión no lineal, ninguno de ellos puede ser considerado el "mejor", en el sentido de la resolución, quizá se puede hablar de eficiencia en cuanto a rapidez de convergencia

pero se descuidan otros aspectos, sucedería algo semejante, si se hablara de sofisticación matemática. En la práctica existe la evidencia documental de que algunos métodos han funcionado satisfactoriamente en ciertos problemas, pero se ha dado el caso de la construcción de problemas particulares, que han derrotado algún método particular, sin embargo podemos aseverar de manera general, que dado un cierto problema, se puede sugerir un método particular, que con adecuadas modificaciones y adecuaciones, se puede lograr una eficiente y rápida convergencia.

En opinión de los autores Draper & Smith -cuyo texto es la base del presente trabajo- el método de Marquardt ha trabajado eficientemente en variadas circunstancias por lo que se considera su selección, una recomendable y práctica decisión. En nuestro parecer, el presente trabajo, hubiese sido más completo si contuviera una sección adicional referida a criterios de selección de métodos y/o criterios de selección de puntos de inicio, pero por cuestiones de tiempo y otros recursos (bibliográficos, sobre todo), se decidió no incluirlas. No obstante podemos mencionar muy amplia o generalmente, que los criterios de selección, para algún método, van a depender mucho, de la experiencia con que cuenta el investigador, sobre estos temas, experiencia práctica mínimamente; por otro lado, significa lo anterior la necesidad de una constante actualización, para mantenerse al tanto del perfeccionamiento o mejoría de los métodos ya existentes, así como también del conocimiento de las nuevas técnicas -que ha nuestro parecer- habrán de surgir.

Es así que se ha llegado al final de ésta sección y por consiguiente del trabajo mismo, finalmente, no resta más que expresar nuestro agradecimiento a algún lector por la atención que le haya podido prestar a las presentes notas. Gracias.

--- FIN ---

BIBLIOGRAFIA.

Bibliografía

Libros:

- 1.- Applied Regression Analysis.
N.R. Draper & H. Smith
J. Wiley and Sons, Inc.; 2nd. Edition, 1981.
- 2.- Optimization Techniques with FORTRAN.
J.L. Kuester & J.H. Mize
Mc. Graw-Hill Book Company, 1973.
- 3.- Forecasting and Time Series Analysis
D.C. Montgomery & L.A. Johnson
Mc. Graw-Hill Book Company, 1976.
- 4.- Métodos Estadísticos.
G.W. Snedecor y W.G. Cochran
C.E.C.S.A., 1978.
- 5.- Time Series Analysis, Forecasting and Control.
G.E.P. Box & G.M. Jenkins
Holden-Day, 1970.
- 6.- Métodos y Modelos de Investigación de operaciones.
Volumen I.
Juán Prawda.
LIMUSA, 1982.

7.- Linear Algebra and Multidimensional Geometry.

N. V. Efimov & E. R. Rozendorn

Mir Publishers, Moscow, 1975.

8.- VAX-11 FORTRAN.

Language Reference Manual

DIGITAL EQUIPMENT CORPORATION, 1978.

Artículos:

9.- An Algorithm for Least Square Estimation

of Non-linear Parameters.

D.W. Marquardt

Journal of Society for Industrial and

Applied Mathematics, 2 ; 1963.

10.- A Method for Minimizing a Sum of Squares

of Non-linear Functions Without Calculating

Derivatives.

M.J.D. Powell.

Computer Journal, 7 ; 1965.

