



ARTÍCULOS

Matemática y computadores

Emilio Machado

Revista de Economía y Estadística, Tercera Época, Vol. 11, No. 1-2 (1967): 1º y 2º Trimestre, pp. 111-122.

<http://revistas.unc.edu.ar/index.php/REyE/article/view/3623>



La Revista de Economía y Estadística, se edita desde el año 1939. Es una publicación semestral del Instituto de Economía y Finanzas (IEF), Facultad de Ciencias Económicas, Universidad Nacional de Córdoba, Av. Valparaíso s/n, Ciudad Universitaria. X5000HRV, Córdoba, Argentina.

Teléfono: 00 - 54 - 351 - 4437300 interno 253.

Contacto: rev_eco_estad@eco.unc.edu.ar

Dirección web <http://revistas.unc.edu.ar/index.php/REyE/index>

Cómo citar este documento:

Machado, E. (1967). Matemática y computadores. *Revista de Economía y Estadística*, Tercera Época, Vol. 11, No. 1-2: 1º y 2º Trimestre, pp. 111-122.

Disponible en: [<http://revistas.unc.edu.ar/index.php/REyE/article/view/3623>](http://revistas.unc.edu.ar/index.php/REyE/article/view/3623)

El Portal de Revistas de la Universidad Nacional de Córdoba es un espacio destinado a la difusión de las investigaciones realizadas por los miembros de la Universidad y a los contenidos académicos y culturales desarrollados en las revistas electrónicas de la Universidad Nacional de Córdoba. Considerando que la Ciencia es un recurso público, es que la Universidad ofrece a toda la comunidad, el acceso libre de su producción científica, académica y cultural.

<http://revistas.unc.edu.ar/index.php/index>

MATEMATICAS Y COMPUTADORES

EMILIO A. M. MACHADO

Frente a todo problema de cálculo es frecuente perder la idea fundamental del proceso, cual es su razonamiento lógico como estructura; y dejarse llevar por los simples cálculos aritméticos o algebraicos que conducen al resultado buscado. En la práctica profesional, aquellas personas que tengan que tratar con la realización efectiva de procesos numéricos necesitarán fundamentalmente saber programar dichos procesos en la forma más racional y económica posible, ya que la realización del mismo correrá por cuenta de un calculista si el volumen no es muy significativo o por cuenta de un computador o conjunto de tratamiento de la información en el caso contrario. En cualquiera de los dos casos será necesario programar el trabajo en un lenguaje accesible al calculista o a la máquina. Trataremos en este trabajo de mostrar la total unidad de las ideas madres del proceso de cálculo y cómo su esencia se aplica tanto en la programación para el calculista como para la máquina electrónica.

I. Naturaleza de un proceso de cálculo

Para entender mejor la esencia de un proceso de cálculo, comenzaremos por estudiarlo en dos ejemplos típicos, independientemente de la forma de realizarlos.

Supongamos primeramente que queremos ajustar un conjunto de puntos cuyas coordenadas son valores correspondientes de un cierto fenómeno cuya relación funcional se quiere determinar. Es

sabido que el método de los cuadrados mínimos no determina el tipo de curva que corresponde a los puntos observacionales, sino que supuesta una determinada estructura, por ejemplo parabólica, determinará de dicha familia la que corresponde mejor a los datos.

El fundamento del método consiste en hacer mínima la suma de los cuadrados de las distancias verticales entre los valores observados de las ordenadas que llamamos y_i y los estimados sobre la curva que se obtendrán calculándolos mediante su ecuación para el correspondiente valor de x para cada y_i medido.

Debe entonces ser mínima la suma

$$\sum_{i=1}^N [y_i - (ax_i^2 + bx_i + c)]^2$$

Teniendo en cuenta que siendo aquí las incógnitas a determinar a , b , c , deberán ser nulas las derivadas parciales de dicha suma con respecto a ellas; es decir, tendremos

$$\begin{aligned} [\sum x_i^4] a + [\sum x_i^3] b + [\sum x_i^2] c &= [\sum x_i^2 y_i] \\ [\sum x_i^3] a + [\sum x_i^2] b + [\sum x_i] c &= [\sum x_i y_i] \\ [\sum x_i^2] a + [\sum x_i] b + N c &= [\sum y_i] \end{aligned}$$

o sea un sistema de Cramer de tres ecuaciones con tres incógnitas, en el cual los coeficientes son sumatorias que se deben calcular en función de las coordenadas de los puntos a ajustar. Para realizar dichos cálculos podemos diseñar un formulario y dar las correspondientes instrucciones para su llenado, de manera que el calculista tenga todo *programado*. En esencia para este sistema de Cramer podemos establecer un *programa* como sigue

x_i	y_i	x_i^2	x_i^3	x_i^4	$x_i y_i$	$x_i^2 y_i$
(1)	(2)	(3)	(4)	(5)	(6)	(7)

En la columna (1) se colocan las abscisas de los puntos dados y en la columna (2) las ordenadas; la columna (3) se obtiene multiplicando cada número de (1) por sí mismo. Lo que podemos indicar:

$$(1) \cdot (1) \longrightarrow (3)$$

La columna (4) se obtendrá por la instrucción:

$$(1) \cdot (3) \longrightarrow (4)$$

es decir, multiplicando cada elemento de la columna (1) por el correspondiente de la columna (3); la columna (5) se obtendrá indistintamente por cualquiera de las siguientes instrucciones:

$$(3) \cdot (3) \longrightarrow (5)$$

$$(1) \cdot (4) \longrightarrow (5)$$

Ambas forman la 4ª potencia de las x_i .

Las dos instrucciones faltantes para completar el cuadro serán:

$$(1) \cdot (2) \longrightarrow (6)$$

$$(2) \cdot (3) \longrightarrow (7)$$

Siguiendo estas instrucciones que se refieren a operaciones entre columnas, que podemos por comodidad llamar memorias, y *que son independientes de los números en sí*, hemos llenado el cuadro; sumando ahora las columnas y teniendo en cuenta que N es el número total de puntos considerados, tendremos calculados los coeficientes del sistema lineal planteado.

Consideremos ahora el otro ejemplo que nos va a permitir planear un *conjunto de instrucciones de cálculo o programa*, de naturaleza un poco diferente al anterior.

Se sabe que la sucesión:

$$X_{n+1} = \frac{X_n + \frac{a}{X_n}}{2}$$

permite calcular \sqrt{a} por un proceso de iteración.

Si queremos hacer un formulario para un calculista y formar el correspondiente conjunto de instrucciones o programa, no debemos olvidar que este proceso iterativo deberá terminarse cuando se tenga la precisión deseada para el valor de \sqrt{a} . Evidentemente el proceso comienza con un primer valor x_0 arbitrario que nosotros daremos, ya escrito, en una columna o memoria

a	x_0	$\frac{a}{x}$	$X + \frac{a}{x}$	$\frac{X + \frac{a}{x}}{2}$
(1)	(2)	(3)	(4)	(5)

La primera instrucción para el llenado del formulario, es decir, del proceso de cálculo, será: ponga x_0 en la columna (2), que nosotros escribiremos:

$$(2) = x_0$$

La segunda instrucción que nos permite llenar la primera línea de la columna (3) y que luego se repetirá, es:

$$(1) : (2) \longrightarrow (3)$$

que dice: el contenido de la columna (1) dividido por el de la columna (2) colóquelo en (3).

A continuación daremos las instrucciones:

$$(2) + (3) \longrightarrow (4)$$

$$(4) : 2 \longrightarrow (5)$$

Con respecto a esta última, dice que el contenido de la columna (4), cualquiera que éste sea, y esto cada vez que se produzca un nuevo contenido, debe dividirse por el número 2 y el resultado se

coloca en la columna (5). Finalmente el valor obtenido es una mejor aproximación de \sqrt{a} que se llevará a la columna (2) y con este valor $x = x_1$ como nueva aproximación se reitera el proceso; el traslado de columna se indicará con:

$$(5) \longrightarrow (2)$$

tendremos entonces sucesivamente en columna (2) las aproximaciones de \sqrt{a} . El proceso se detendrá cuando tengamos calculada la cifra decimal exacta requerida; esto puede identificarse por el hecho de que la diferencia entre dos aproximaciones sucesivas en valor absoluto sea menor que el error admitido, en cuyo caso el proceso ha terminado; en caso contrario, se continúa hasta lograrlo.

Esta instrucción podemos escribirla:

$$| X_{n+1} - X_n | < \varepsilon ? \begin{cases} \text{si : fin} \\ \text{no : continúa} \end{cases}$$

En estos dos ejemplos, hemos querido hacer notar que la realización de los problemas en ellos propuestos, exigen el desarrollo de un programa de cálculo formado por un conjunto de instrucciones que operan sobre las columnas o memorias; dicho programa será independiente de los números en sí y constituye la estructura lógica del proceso de cálculo. El primer ejemplo es general para todo ajuste de una parábola por mínimos cuadrados y el segundo para el cálculo de la raíz cuadrada de un número a .

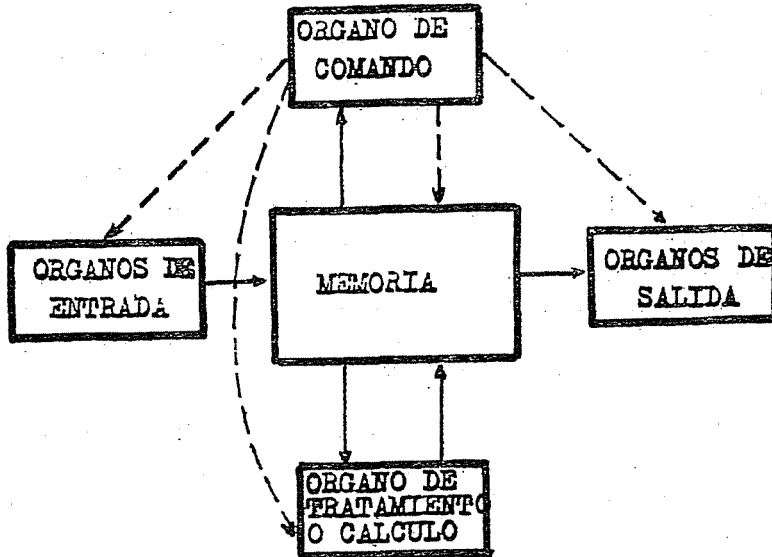
II. *Constitución de un computador o conjunto de tratamiento de la información*

Antes de pasar al análisis de la constitución lógica de un conjunto de tratamiento de la información, pensemos en la estructura del "conjunto" formado por un calculista, una máquina de escritorio, papel y lápiz y eventualmente algunas tablas.

Evidentemente el calculista debe recibir un "conjunto de instrucciones" que constituyen el programa de cálculo; en el caso del ajuste de una curva por el método de los mínimos cuadrados no interesa la deducción analítica del valor de las constantes, sino las operaciones a realizar sobre los números experimentales, es decir que en este caso el programa de cálculo estará formado por las instrucciones que recibe el calculista: "calcule para cada valor de x su cuadrado, su cubo, su cuarta potencia", etc.; cuando el calculista por sus órganos de entrada ha recibido el programa, constituido por las instrucciones de cálculo a efectuar sobre los números datos, lo registra en forma de cuadro en un papel, luego recibe los datos del problema, es decir los números sobre los cuales debe desarrollar el programa; estos datos, a los que anota en las columnas correspondientes del papel, constituyen conjuntamente con el programa también anotado en el papel toda la información necesaria. Una vez en poder de la información el calculista efectúa en la máquina de calcular las operaciones indicadas en el programa y va anotando los resultados parciales y finales en la hoja de papel; eventualmente puede hacer uso de tablas auxiliares para el cálculo. Una vez terminado entrega los resultados. Es claro que en cualquier momento podemos pedirle al calculista un resultado parcial que nos lo entregará una vez logrado.

La misma estructura lógica tiene un computador o conjunto de tratamiento de la información; ésta entra mediante los órganos de entrada, lector de tarjeta perforada, lector de banda perforada, etc.; la información así recibida, ya sean instrucciones o datos se registran en una memoria, el programa actúa mediante el órgano de comando sobre el órgano de cálculo que realiza las operaciones indicadas, los resultados de las mismas se van registrando en memorias hasta terminar el proceso, los resultados finales se sacan de la memoria y se entregan al interesado en forma impresa, en forma de banda o tarjeta perforada, etc., mediante los órganos de salida. El computador puede tener registrado para su oportuno empleo, tablas necesarias para el cálculo.

Podemos, de acuerdo con lo dicho, esquematizar un conjunto de tratamiento de la información, en la siguiente forma



III. Representación de la información

El problema de la codificación de la información y su tratamiento exige el uso de sistemas de numeración adecuados y que permitirán la comunicación entre el hombre y la máquina.

Un sistema de numeración está caracterizado por su base b , que es la cantidad de símbolos diferentes utilizados para escribir los números en el sistema considerado. El sistema habitual es el decimal de base 10. Los diez símbolos son las cifras 0 a 9. Un número se escribe entonces:

$$N' = a_0 + a_1 10 + a_2 10^2 + \dots + a_n 10^n$$

donde las a_i son una de las diez cifras decimales. Por comodidad al escribir un número en el sistema decimal nosotros lo hacemos sobreentendiendo las potencias de 10, en la forma

$$N = a_n a_{n-1} a_{n-2} \dots a_1 a_0$$

Sería ciertamente más normal hacerlo en el orden inverso

$$N = a_0 a_1 a_2 \dots a_n$$

Nuestra manera de escribir los números, contraria a las reglas lógicas del álgebra, tiene su origen en el hecho de que hemos copiado el sistema árabe sin tener en cuenta que ellos escriben de derecha a izquierda.

En general en un sistema en base b un número N se escribirá

$$N = a_0 b^0 + a_1 b^1 + a_2 b^2 + \dots + a_n b^n$$

donde cada a_i es uno de los b símbolos que constituyen el sistema de numeración en base b . Razones tecnológicas nos han llevado a utilizar en los conjuntos de tratamiento de la información el sistema binario puro y algunos derivados.

Daremos a continuación ideas sobre estos sistemas, dejando en claro que si bien los sistemas de tratamiento de la información trabajan en ellos, la comunicación con el exterior se realiza en el sistema decimal, ya que el computador convierte de por sí decimal a binario y recíprocamente.

En los sistemas binarios disponemos de sólo dos símbolos 0 y 1; por lo tanto un número estará constituido por un conjunto de ceros y unos.

Sea por ejemplo el número binario 1001101011. Lógicamente nos es imposible reconocer a simple vista su magnitud en forma decimal. A éste, nuestro sistema, debemos convertirlo. Para ello, el método consiste en escribir debajo de cada cifra binaria la potencia de 2 que le corresponde, es decir, el rango binario; luego, sumar las potencias correspondientes a las cifras unos. En nuestro caso tendremos:

HOMENAJE AL DR. JOSE YOCCA

$$\begin{array}{cccccccccc}
 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
 512 & 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
 512 + 0 + 0 + 64 + 32 + 0 + 8 + 0 + 2 + 1 = 619
 \end{array}$$

La transformación inversa, es decir, decimal a binaria, es más delicada. Para realizarla es necesario ir dividiendo sucesivamente el número a transformar y sus cocientes por dos. La sucesión formada por los restos nos dará el número escrito en base dos a partir de las potencias menores 2^9 a la mayor que resulte. Para nuestro número 619 tendremos:

$$\begin{array}{l}
 619/2 / 1 \\
 309/2 / 1 \\
 154/2 / 0 \\
 77/2 / 1 \\
 38/2 / 0 \\
 19/2 / 1 \\
 9/2 / 1 \\
 4/2 / 0 \\
 2/2 / 0 \\
 1/2 / 1
 \end{array}$$

Lo que nos da:

1001101011

Análogamente si consideramos el número 263 tendremos:

$$\begin{array}{l}
 263 / 2 / 1 \\
 131 / 2 / 1 \\
 65 / 2 / 1 \\
 32 / 2 / 0 \\
 16 / 2 / 0 \\
 8 / 2 / 0 \\
 4 / 2 / 0 \\
 2 / 2 / 0 \\
 1 / 2 / 1
 \end{array}$$

Es decir, tendremos:

10000111

En cuanto a las operaciones en binario, son sumamente simples, si bien no rápidas, a causa de la gran cantidad de cifras necesarias para representar un número.

Las operaciones de suma y producto responden a las siguientes tablas:

+	0	1
0	0	1
1	1	10

X	0	1
0	0	0
1	0	1

Ejemplo: sea la suma $263 + 619$

$$\begin{array}{r}
 263 = 1\ 00000\ 111 \\
 619 = 10\ 01101\ 011 \\
 \hline
 11\ 01110\ 010
 \end{array}$$

Este número corresponde a:

$$512 + 256 + 64 + 32 + 16 + 2 = 882$$

Las operaciones de diferencia, producto y cociente son también sencillas y las dejamos por cuenta del lector.

Si el sistema binario es económico desde el punto de vista del número de símbolos, no lo es en función de la longitud de los números. Cuando se utiliza el sistema binario las operaciones aritméticas son más rápidas a pesar de la mayor longitud de los operandos. Por el contrario, la transformación binario decimal y recíproca son relativamente largas; consecuencia de esto es que cuando la cantidad de datos y de resultados son importantes el tiempo necesario para la traducción en la máquina es considerable. Para mantener las ventajas del sistema binario y eliminar en lo posible los inconvenientes es que se ha considerado utilizar códigos más fáciles de traducir en decimal. El código clásico es el Decimal Codificado Binario (DCB). Se utilizan en este código cuatro cifras binarias para representar un número decimal; cada cifra binaria corresponde a una potencia de dos. Tenemos así los valores 1, 2, 4, 8; se dispone entonces de 16 combinaciones posibles, es decir que podemos representar 16 símbolos diferentes. Este código necesita más símbolos que el binario puro.

La tabla de transformación Decimal —DCB será la siguiente:

<i>Decimal</i>	<i>D C B</i>
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Ahora la traducción Decimal a DCB y viceversa son sencillas.
 Por ejemplo el número decimal 273 se escribirá en DCB

0010 , 0111 , 0011

En cuanto a la tabla de sumar habrá una pequeña modificación a realizar en la binaria pura; ésta se debe a que si se sumaran como binarios dos DCB, al sumar las cuartas posiciones binarias, si se produce un salto, éste equivale a 16 y no a 10 como en DCB. Para mejor comprender tomemos un ejemplo:

Sea la suma:

7 3	en DCB	0 1 1 1	0 0 1 1
3 1	„	0 0 1 1	0 0 0 1
<hr/>		<hr/>	<hr/>
1 0 4	en DCB	1 0 1 0	0 1 0 0

El número 1 0 1 0 no existe en DCB, ya que es el número 10 en binario puro, pero este número 10 en DCB se escribe mediante dos grupos de cuatro cifras en la forma 0001 0000 que leído en binario puro corresponde a 16; para darle sentido al símbolo 1010 en DCB, en las operaciones en que aparezca se debe sumar la diferencia 6 entre los saltos del Decimal Codificado Binario y el Bi-

nario Puro; esto es, el número 0110 al símbolo que carezca de sentido. De acuerdo con lo dicho nuestra operación quedará:

73	0 1 1 1	0 0 1 1
31	0 0 1 1	0 0 0 1
	1 0 1 0	0 1 0 0
6	0 1 1 0	
0 0 0 1	0 0 0 0	0 1 0 0
1	0	4

Dejamos de lado las otras operaciones de las cuales la resta se realiza llevando el complemento a nueve de uno de los operandos, la multiplicación por reiteración de suma y la división por procesos análogos. Es necesario hacer notar que en los conjuntos electrónicos, el cero decimal no se codifica 0000 en razón de que este símbolo puede confundirse con la falta de información. Es lógico que se utilice entonces uno de los símbolos sobrantes, por ejemplo 1010, que es equivalente a 10 en binario puro.

Para la representación de la información no numérica, es claro que se podrán utilizar ya sea los símbolos que no representan números en la DCB que no son suficientes en cantidad o bien otra codificación que no modifique la esencia de la numérica y que permita contar con mayores posibilidades para la codificación de letras y símbolos operatorios y lógicos. Para esto se agregan adelante de las cuatro cifras necesarias para la representación numérica otras dos cifras que serán siempre ceros en el caso de los números. Así por ejemplo, tendremos:

3	0 0	0 0 1 1
7	0 0	0 1 1 1
8	0 0	1 0 0 0

Tendremos ahora $16 \times 4 = 64$ símbolos posibles, diez ocupados por cifras y el resto para la restante información codificada.

Saldría del objetivo de este trabajo dar más detalles sobre la codificación de la información. Debemos, eso sí, decir que la introducción de la información en un conjunto de tratamiento se realiza ya sea mediante tarjetas perforadas y/o banda perforada en general.

IV. *El C. A. H.*

En este párrafo “construiremos” un *Computador Automático Hipotético* y aprenderemos a programarlo. Aun cuando dicho computador no exista, la habilidad en su programación se traslada de inmediato a cualquier computador o conjunto de tratamiento de la información existente.

Comencemos con una situación totalmente abstracta. Imaginemos un número ilimitado de compartimentos o celdas, numeradas a partir del cero y en cada una de las cuales puede almacenarse un número real o una instrucción. Los números asignados a las celdas se llaman las “direcciones” de las mismas o de las “memorias”. Hacemos la convención siguiente: si α es la dirección de la celda, entonces (α) indica el contenido de la celda. Análogamente si X es un número real almacenado en una celda o memoria $[X]$ indica la dirección de la celda que contiene X . Es $([X]) = X$

Nótese que (α) está unívocamente definido pero no así $[X]$ puesto que más de una celda o memoria puede contener X .

Definamos ahora algunas instrucciones de C.A.H. Tengamos para ello en cuenta lo dicho con respecto al contenido de una memoria. En ellas cabe o bien un número real o una instrucción; esto, que no es así exactamente en la realidad, es una propiedad simplificativa de nuestro C.A.H. pero que no invalida el razonamiento.

Usaremos en lo que sigue el símbolo \rightarrow para indicar: “ir a”, “vaya a”, “reemplace”, cualquiera de estas expresiones.

Entonces unas instrucciones de las siguientes expresiones:

$$\begin{aligned} (\alpha) + (\beta) &\rightarrow \gamma \\ (\alpha) - (\beta) &\rightarrow \gamma \\ (\alpha) \cdot (\beta) &\rightarrow \gamma \\ (\alpha) \div (\beta) &\rightarrow \gamma \\ (\alpha) < (\beta) &\rightarrow \gamma \end{aligned}$$

Todas, excepto la última, son instrucciones aritméticas; la primera significa que los contenidos de las celdas α y β se suman y el resultado se deposita en la memoria γ . Análoga interpretación puede darse a las tres siguientes instrucciones. Es fundamental suponer que una vez realizada una instrucción por la máquina se respeta el orden del programa, es decir que se efectúan sucesivamente escritas.

La última instrucción, sin embargo, es de una naturaleza diferente; es una instrucción *lógica*. Aquí los contenidos de α y β son comparados y si $(\alpha) < (\beta)$ la próxima instrucción realizada es la contenida en la celda γ y no la que sigue en el programa; mientras que si $(\alpha) \geq (\beta)$, la instrucción siguiente es la que sigue inmediatamente en el programa. En otras palabras, cuando se cumple la condición de la instrucción lógica se produce un salto.

Estamos ahora en condiciones de escribir un programa simple que nos permitirá entender mejor la esencia del problema.

Las instrucciones se escriben con su dirección a la izquierda y su explicación a la derecha. Así, $10 (4) + (8) \rightarrow 4$ significa que en la celda o memoria 10 está registrada la instrucción $(4) + (8) \rightarrow 4$

Sea, entonces, la programación del cálculo de x^p , donde x es un número real y p un entero positivo.

Pongamos:

- | | |
|-----------|----------------------------------|
| $p = (1)$ | (p contenido de la memoria 1) |
| $x = (2)$ | (x contenido de la memoria 2) |
| $1 = (3)$ | (1 contenido de la memoria 3) |

Nuestro programa será:

- 10.— $(4) - (4) \rightarrow 4$ (vacía la memoria 4)
- 11.— $(3) \cdot (3) \rightarrow 5$ (contenido de la memoria 5 igual a 1,)
- 12.— $(3) + (4) \rightarrow 4$ ($1 + r$ reemplaza a r)
- 13.— $(2) \cdot (5) \rightarrow 5$ ($X \cdot X^r = X^{1+r}$ reemplaza a X^r)
- 14.— $(4) < (1) \rightarrow 12$ (si $r < p$ vaya a la instrucción 12, en caso contrario a la siguiente, es decir la 15)
- 15.— FIN

Para mejor estudiar este programa y cualquier otro, es conveniente dibujar un conjunto de celdas que representarán las memorias, ubicar en ellas las instrucciones de base y luego ir viendo el movimiento como consecuencia de las sucesivas instrucciones del programa.

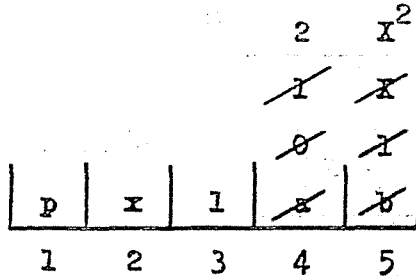
En el programa aquí planteado tenemos cinco memorias en las cuales se produce el proceso de cálculo, inicialmente p en la memoria 1, x en la 2, 1 en la 3; cualquier valor en 4 y 5.

La instrucción de la memoria 10 vacía el contenido de la memoria 4

La 11 pone 1 en la memoria 5

La 12 pone 1 en la memoria 4

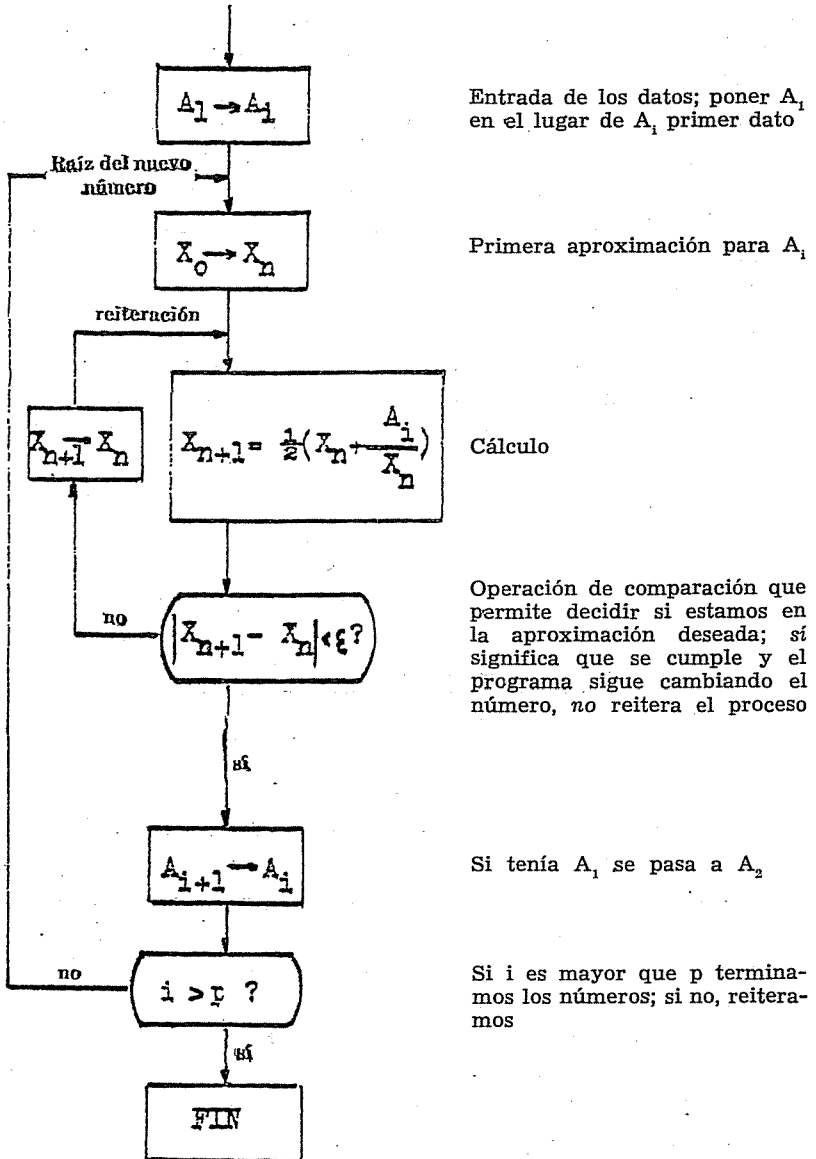
La 13 pone x en la memoria 5, la primera vez, y al repetirse va formando las sucesivas potencias de x como consecuencia de la instrucción 14.



El análisis mediante celdas dado más arriba es el método ideal para ver el desarrollo de un programa. Por el contrario, si queremos realizar efectivamente un programa en base a un planteo matemático, debe comenzarse por el "organigrama" del problema a analizar; éste es la graficación de las operaciones que constituyen el cálculo en su sucesión lógica.

Consideremos el caso ya tratado al principio de este trabajo del cálculo de la raíz cuadrada, pero en vez de considerar un solo número cuya raíz cuadrada queremos determinar, tomemos una sucesión finita de valores A_i , con $i = 1, 2, 3, \dots, p$.

Este proceso puede esquematizarse mediante el siguiente organigrama:



Evidentemente el organigrama da una idea gráfica del conjunto de las operaciones. En base a él se deberá construir el programa correspondiente.

Dejando de lado este problema que puede resolverse fácilmente, insistiremos una vez más que en el proceso de cálculo, en este caso considerado como en cualquier otro, lo fundamental es la estructura lógica, indicada en el organigrama y desarrollada en el programa.

Para terminar consideremos el problema de la multiplicación de dos matrices, una matriz A de orden $m \times p$ y otra B de orden $p \times n$. Simbólicamente, a dicho producto podemos escribirlo

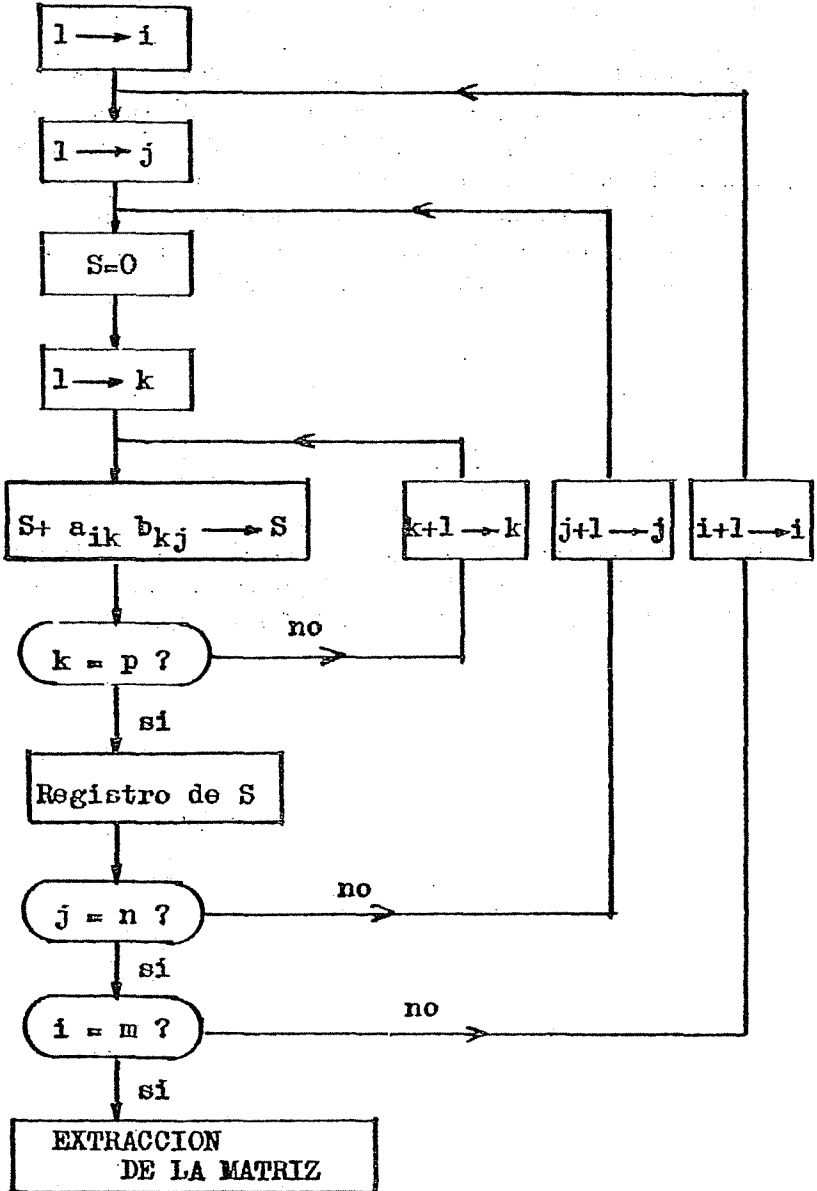
$$C = A \cdot B$$

$$(m \times n) \quad (m \times p) \quad (p \times n)$$

$$C = [c_{ij}] \quad c_{ij} = \sum_{k=1}^p a_{ik} b_{kj} \quad \begin{array}{l} i = 1, 2, 3, \dots, m \\ j = 1, 2, 3, \dots, n \end{array}$$

El organigrama de cálculo supone la matriz A ordenada por filas y la B ordenada por columnas. El proceso diagramado permite calcular los elementos de la matriz C fila por fila.

El organigrama correspondiente es el siguiente



Como dijimos al comienzo de este trabajo, hemos tratado de mostrar la unicidad del concepto básico de programa de cálculo y su total independencia del método de realización. Hemos visto cómo las mismas ideas que sirven de base al cálculo manual son las que se usan en el cálculo mediante computadores y cómo las modificaciones son simples problemas de codificación*.

Todo proceso de cálculo es un proceso de tratamiento de la información y ésta es la esencia misma en la que se debe insistir y no el método mediante el cual se la realiza, que es de carácter secundario. La teoría de la información y la cibernética van a aquella esencia misma del conocimiento científico y no pueden ser dejadas de lado cuando se quieren enseñar o estudiar con rigor y simplicidad los problemas del análisis numérico tan comunes en la ciencia y técnica contemporáneas.

* Esto vale aun para los lenguajes más evolucionados como el Algol, Fortran, etc., en donde todo el problema se reduce a establecer lo que se llama un compilador que traduce el algoritmo programa en lenguaje máquina.