

## MECANIZACION DE CALCULO NUMERICO

Por FEDERICO ALICART GARCES

Ingeniero de Caminos

Subdirector del Gabinete de Cálculo del Centro de Estudios y Experimentación de Obras Públicas

*Se pone de manifiesto con algunos ejemplos el impacto de los computadores electrónicos en los métodos de cálculo que utilizaba el ingeniero de antaño. También se alude al problema de las estructuras gramaticales de los modernos lenguajes automáticos y la conveniencia de que nuestros computadores acepten el castellano.*

El tratamiento de la información puede compararse a un proceso en cadena, uno de cuyos eslabones es el computador electrónico.

Al principio de la cadena hay un órgano receptor por donde la información penetra en el sistema. Es la información de entrada o "input".

Al final de la cadena hay otro órgano que nos da los resultados apetecidos. Es la información de salida o "output".

Se repiten con tanta frecuencia en informática los conceptos "input" y "output" que sería deseable encontrar dos voces españolas equivalentes y dignas de merecer la aceptación unánime de nuestros técnicos.

El esquema más sencillo de esta cadena sería el representado en la figura 1.

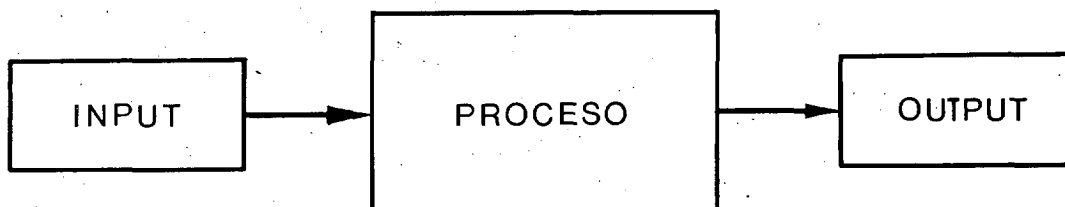


Figura 1.

El recuadro central de este esquema se refiere a las partes más esenciales de la cadena que comprenden multitud de canales para absorber el flujo informativo y transmitirlo a otros canales después de atravesar registros, órganos de control, circuitos lógicos, etc.

No hay inconveniente en modificar el esquema anterior, sustituyendo la palabra proceso por la palabra computador y poniendo de manifiesto junto a la unidad cen-

tral, representativa del núcleo del sistema, otros recuadros más pequeños para destacar las unidades periféricas de entrada y salida (fig. 2).

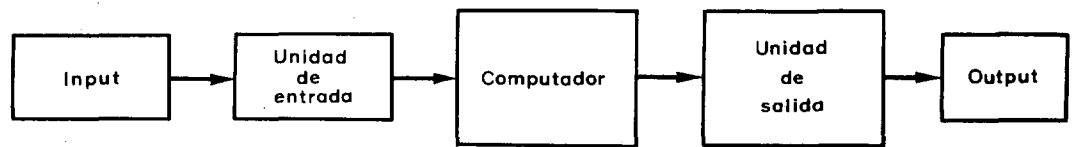


Figura 2.

Pero un sistema concebido de esta manera tendría una acción muy limitada, pues sólo sería capaz de realizar aquel repertorio de procesos que tuvo en su mente el ingeniero proyectista.

Vamos a referirnos concretamente a los computadores dígito-universales, capaces de adaptarse a las condiciones de cualquier algoritmo lógico-matemático y que en esencia se reducen a tres operadores elementales

y, o (no exclusivo), no, que el lenguaje ALGOL represente mediante los símbolos



Este computador universal, capaz de admitir los más variados procesos requiere, en cada caso, ser debidamente instruido antes de desencadenar el proceso.

Hay que suministrarle un programa o lista de instrucciones que sucesivamente deben ser realizadas para convertir el "input" en el "output" deseado.

Además, ese programa debe estar codificado en un lenguaje aceptable por la máquina que, en realidad, no es otro que el propio lenguaje máquina, es decir, el de nuestra máquina, aquella que manejamos, porque cada tipo de máquina tiene su lenguaje individual de comunicación con el mundo exterior.

Este problema de los lenguajes creó verdaderos obstáculos a la difusión de los computadores durante el primer decenio de la era informática (aproximadamente de 1945 a 1954) hasta que surgió la idea genial de los lenguajes simbólicos (FORTRAN, ALGOL, COBOL, BASIC, MAD, JOVIAL, LISP, CPL, etc.), o sea la llamada programación automática que hoy conocen y aplican todos los profesionales.

Entonces nos encontramos con la figura 3, donde, para mayor claridad, hemos colocado dos canales que representan separadamente la entrada de datos y la entrada del programa.

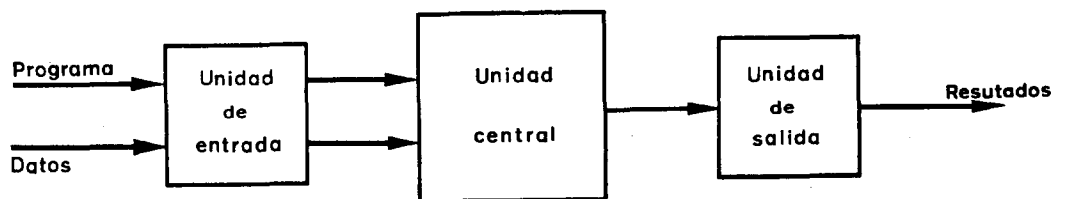


Figura 3.

En los primeros tiempos había que preparar cuidadosamente cada programa, pues se trataba de penetrar en campo inexplorado.

Redactar un buen programa podía ser tarea de horas, de semanas, de meses o aun de años, según las dificultades del algoritmo.

En el momento actual, al cabo de cuatro lustros de incesantes progresos, cualquier Centro o Gabinete de Cálculo dispone de centenares o incluso millares de programas perfectamente adaptados a las máquinas que utiliza y puede hacer frente, sin esfuerzo ni demora, a la ejecución mecanizada de cualquier trabajo ordinario.

Hay programas repetitivos, como el cálculo de estructuras reticulares, que son de enorme productividad, pero un Gabinete siempre dispone de elementos para programar nuevos problemas reservados y particulares de un determinado cliente. Aunque en este caso el costo de la programación sea relativamente elevado, cabe la posibilidad de reducir esta carga económica en nuevas repeticiones del mismo programa con otros datos al servicio del mismo cliente.

La técnica de la programación ha realizado avances increíbles en poco tiempo, que son pasos muy significativos hacia la creación de la inteligencia artificial.

Se trata de que el mismo computador sea capaz de modificar un programa básico con arreglo a determinados criterios y teniendo en cuenta los resultados prácticos que vayan obteniéndose en sucesivas aplicaciones del mismo.

Es un proceso similar a la educación de un niño dócil que acepta los dictados de sus educadores, conforme a ciertas leyes físicas o morales que la sociedad establece.

El esquema 4 explica con claridad estas ideas:

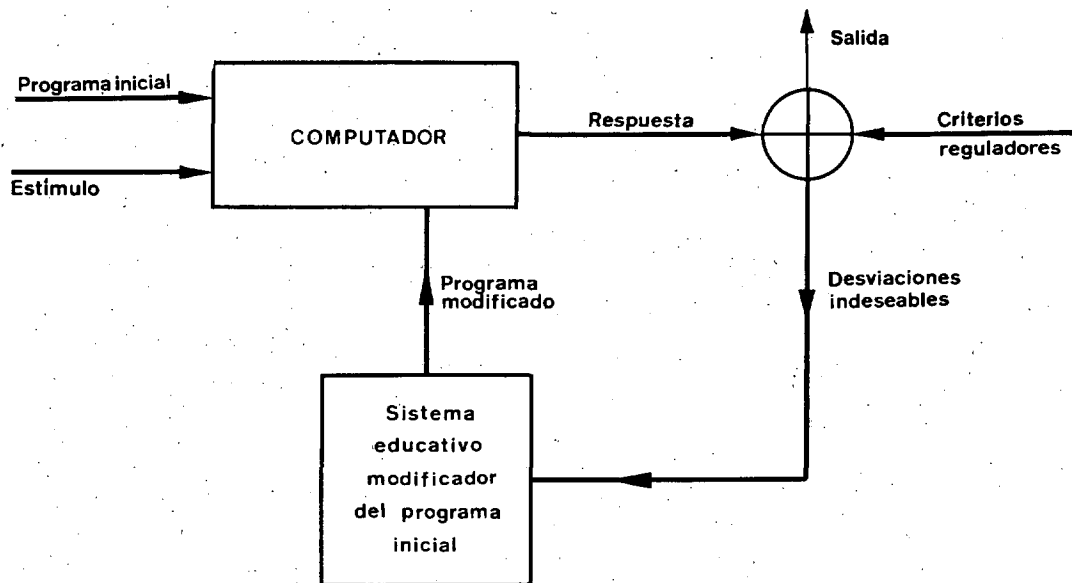


Figura 4.

Así acontece con el computador que juega al ajedrez no con el rey y el roque estudiado por Torres Quevedo, sino con sus piezas completas y con la experiencia adquirida a través de los sucesivos juegos, llega a depurar su estrategia hasta vencer al mismo programador, a su primer maestro, a quien siempre superó en memoria y reflejos.

No hace falta advertir que desde la primera línea de este artículo estamos hablando de procesos mentales que el computador mecaniza, pero en particular nos interesa decir algo concreto sobre la mecanización de procesos numéricos.

Las bibliotecas de nuestras Universidades están llenas de métodos matemáticos pertenecientes al siglo XIX, elegantes de forma y rigurosos en su contenido, pero que los ingenieros no utilizan porque no tienen tiempo de hacer tantas operaciones aritméticas y prefieren utilizar recetas más o menos aproximadas y rudimentarias, colocándose en una posición que hoy no puede mantenerse.

Para más claridad nos valdremos de algunos ejemplos:

*Primero.*

La estabilidad de un sistema dinámico está determinada por su respuesta a determinados impulsos o perturbaciones.

Siendo  $W(s)$ ,  $X(s)$ ,  $Y(s)$ , respectivamente, la función transfer del sistema y las transformadas de Laplace del "input" y del "output" sabemos que en el caso de sistemas lineales invariantes respecto al tiempo, la estabilidad depende de la naturaleza de las raíces de la ecuación característica.

$$W(s) = 0$$

Un estudio completo del sistema exigiría resolver esta ecuación, pero en multitud de ocasiones sólo interesa saber si es posible garantizar la estabilidad del mismo y ello depende de que las  $n$  raíces de

$$a_0 s^n + a_1 s^{n-1} + \dots + a_{n-1} s + a_n = 0 \tag{1}$$

tengan sus partes reales menores o iguales a cero.

En caso afirmativo diremos que la ecuación es estable.

En los tratados puede leerse el método que Routh dio a conocer en 1877 para determinar el valor booleano (cierto o falso) de la proposición siguiente:

La ecuación (1) es estable

(2)

No necesitamos saber más.

El algoritmo de Routh consta de sucesivas etapas calculando en cada una de ellas los arreglos  $b [ ]$ ,  $c [ ]$ ,  $d [ ]$ , etc., según indica la adjunta tabla

$a_0$	$a_2$	$a_4$	$a_6$	.....
$a_1$	$a_3$	$a_5$	$a_7$	.....
$b_1$	$b_2$	$b_3$	$b_4$	.....
$c_1$	$c_2$	$c_3$	$c_4$	.....
.....				

siendo

$$b_1 = \frac{a_1 a_2 - a_0 a_3}{a_1} \quad \gg \quad b_2 = \frac{a_1 a_4 - a_0 a_5}{a_1} \quad \gg \quad b_3 = \frac{a_1 a_6 - a_0 a_7}{a_1} \dots\dots$$

$$c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1} \quad \gg \quad c_2 = \frac{b_1 a_5 - a_1 b_3}{b_1} \quad \gg \quad c_3 = \frac{b_1 a_7 - a_1 b_4}{b_1} \dots\dots$$

.....

.....

Este proceso se interrumpe cuando los elementos de una nueva fila sean todos nulos.

Si contamos en ese momento el número de variaciones de signo en la primera columna de la tabla, tendremos el número de raíces con parte real positiva de la ecuación propuesta.

Por tanto, si no hay ninguna variación, el valor de la proposición (2) es CIERTO (True), y el sistema dinámico es ESTABLE.

En cambio, si se produce alguna variación obtendremos el valor FALSO (False), y el sistema será inestable.

Con esto ha terminado el papel del analista y comienza el papel del programador, encargado de explicarle a la máquina todo lo que tiene que hacer.

Prescindiendo de algunas sencillas declaraciones iniciales y de las instrucciones de formato para la entrada y salida de datos, todo el programa preparado en ALGOL 60 consta de las siguientes instrucciones:

**"begin**

estable := false;

for j := 0 step 1 until n-1 do

begin

if a [0] × a [j + 1] ≤ 0 then goto sal;

c := -a [j] / a [j + 1];

for k := j + 2 step 2 until n-1

do a [k] := a [k] + c × a [k + 1]

end;

estable := true;

sal:

**end".**

La ejecución de este programa en máquina es casi instantáneo incluso para altos valores de  $n$ .

Apenas se ha pulsado la tecla *start* ya aparece la respuesta deseada:

true o false

a través de la máquina de escribir situada en consola.

Nuestro problema de cálculo ha quedado perfectamente mecanizado.

Del mismo modo tenemos centenares de algoritmos programados en nuestra biblioteca al servicio de quien lo solicite.

*Segundo.*

Otro proceso de gran interés que tiene programado el Gabinete es el ajuste de una función empírica  $y = f(x)$  mediante los polinomios MIN-MAX que dan la desviación mínima entre todas las máximas correspondientes a polinomios de grado  $\leq n$  en un intervalo  $(a, b)$ .

Chebyshev obtuvo genialmente su método de ajuste en el siglo pasado, pero los cálculos eran de ejecución demasiado penosa.

Hoy tenemos mecanizado ese proceso al servicio de nuestros estadísticos y economistas.

Una exposición del programa será objeto de una publicación especial del Gabinete.

*Tercero.*

Otro programa muy interesante de Cálculo Numérico estudiado por el Gabinete es la aproximación racional de una función  $y = f(x)$ , conociendo  $n$  puntos:

$$(x_1, y_1) \quad (x_2, y_2) \quad \dots \dots \dots \quad (x_n, y_n)$$

Se trata de programar la fórmula

$$y = y_1 + \frac{x - x_1}{\rho_1(x_2, x_1)} + \frac{x - x_2}{\rho_2(x_3, x_1, x_2) - y_1} + \frac{x - x_3}{\rho_3(x_4, x_1, x_2, x_3) - \rho_1(x_1, x_2)} + \frac{x - x_4}{\rho_4(x_5, x_1, x_2, x_3, x_4) - \rho_2(x_1, x_2, x_3)} + x - x_5 + \dots$$

donde las diferencias recíprocas  $\rho$  son funciones de las coordenadas conocidas.

Es asombrosa la facilidad con que se obtienen los valores numéricos que intervienen en el desarrollo, así como los valores interpolados de la función empírica.

Se trata de un recurso de gran utilidad para el ingeniero y el investigador.

## R E S U M E N

1.º Las técnicas de mecanización son relativamente sencillas para los ingenieros especializados.

2.º El usuario no necesita tener conocimientos previos sobre el cálculo electrónico. Le basta consultar con el Gabinete sobre la índole del problema que le interesa.

3.º En fecha no lejana, cuando se instalen en España servicios públicos teleinformáticos, podrá conectarse directamente con el computador central, desde el propio domicilio o despacho del abonado a través de un terminal particular y línea telefónica, para obtener instantáneamente la información deseada sobre programas de interés registrados en una guía general a la que podrán agregarse, con carácter privado o secreto, otros programas que sean propiedad particular del abonado. Todo ello con tarifas a convenir.

4.º Técnicamente errónea y económicamente nociva es la tendencia, que se observa en algunas grandes empresas, a multiplicar pequeños equipos al servicio de pequeñas secciones dentro de la empresa. Los nuevos horizontes del teleproceso y del time-sharing condenan tal derroche de dinero, agravado por las dificultades de formar y retribuir adecuadamente los cuadros de personal altamente especializado en torno de cada máquina, obligados a seguir al día las evoluciones de una técnica avanzadísima.

5.º El ALGOL 60 al igual que cualquier FORTRAN tiene sus símbolos básicos, su sintaxis y su semántica que no podemos modificar en el menor detalle.

Un punto y coma suprimido; escribir "**principio**" o **Begin** en vez de **begin** o incluso poner **falso** en vez de **false**, supondría la paralización del programa, pues la máquina ya no es capaz de interpretar lo que se le pide.

Sería un desatino recurrir a los métodos del aprendizaje y de la inteligencia artificial, con el tiempo formativo que ello implica, para corregir tan nimios detalles.

6.º Algunos países están rechazando la servidumbre de los símbolos básicos en lengua inglesa y redactan los programas en su idioma oficial.

En realidad (caso del ALGOL 60) se trata de los 23 símbolos siguientes:

<b>true</b>	<b>for</b>	<b>begin</b>	<b>switch</b>
<b>false</b>	<b>do</b>	<b>end</b>	<b>procedure</b>
<b>goto</b>	<b>step</b>	<b>real</b>	<b>label</b>
<b>if</b>	<b>until</b>	<b>integer</b>	<b>string</b>
<b>then</b>	<b>while</b>	<b>Boolean</b>	<b>value</b>
<b>else</b>	<b>comment</b>	<b>array</b>	

de los cuales en español podrían conservarse sólo **real** y **Boolean**. Los restantes podrían sustituirse por las formas castellanas

<b>cierto</b>	<b>para</b>	<b>principio</b>	<b>conmutador</b>
<b>falso</b>	<b>hacer</b>	<b>final</b>	<b>proceso</b>
<b>ira</b>	<b>paso</b>	.....	<b>etiqueta</b>
<b>si</b>	<b>hasta</b>	<b>entero</b>	<b>cadena</b>
<b>entonces</b>	<b>mientras</b>	.....	<b>valor</b>
<b>sino</b>	<b>comentario</b>	<b>cuadro</b>	

Finalmente habría que revisar algunos símbolos de las instrucciones I/O, de formato y de los identificadores de funciones standard; por ejemplo, poner **sen** en vez de **sin**, **raíz2** en vez de **sqrt**, **parte entera** en vez de **entier**, **longitud** en vez de **length**, etcétera.

7.º Esta nacionalización del lenguaje automático puede efectuarse modificando los compiladores que ahora se manejan, tanto en ALGOL, como en FORTRAN, en COBOL, en JOVIAL (utilizado por el Ejército americano), etc.

Es tarea relativamente sencilla y los franceses la han emprendido.

Lo que deba realizarse en nuestro país tiene que ser estudiado y decidido muy cuidadosamente por nuestras autoridades competentes.