Boletín de Matemáticas
Vol. XIX Nº 3 (1985) **APUNTES**

UN METODO NUMERICO PARA CALCULAR FUNCIONES BESSEL DE PRIMERA CLASE DE ORDEN ENTERO

Mauricio García

1. INTRODUCCION.

En un reciente artículo, el profesor Takeuchi [1] muestra los peligros al utilizar un computador en la predicción de la convergencia de una sucesión. El "desastre" ocurrido se debió al empleo de una fórmula de recurrencia y a la falta de precisión inherente de la máquina utilizada.

Una situación parecida se encuentra cuando por medio de fórmulas de recurrencia se quiere calcular el valor de una función esférica de Bessel o los valores de las funciones de Bessel de primera clase de orden entero.

En una comunicación previa [2], se mostró cómo, haciendo uso del método de Miller, se obvia el problema de la precisión en las funciones esféricas de Bessel.

En el presente artículo se describe un método numérico para el cálculo de funciones Bessel de primera clase de orden entero, y el programa correspondiente escrito para una calculadora *Casio FX-702-P* la cual emplea un *Basic* restringido.

2. EL PROBLEMA.

La fórmula de recurrencia para las funciones Bessel de orden entero viene dada por la siguiente relación: [3]

$$J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x). \tag{1}$$

Por consiguiente, al conocer $J_o(x)$ y $J_1(x)$, en principio es posible calcular recurrentemente la función para cualquier orden, una vez establecido el argumento.

Por otra parte, la expresión general para las funciones $J_n(x)$ se hace por medio de la serie: [3]

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{n+2k}}{k!\,\Gamma(n+k+1)} \tag{2}$$

con Γ, la función gama.

La convergencia de la serie de la ecuación (2), no es siempre rápida para x grande, y por tal razón, en las tablas donde se tabulan los $J_n(x)$ tan sólo aparecen las órdenes más bajos, en general $J_o(x)$ y $J_1(x)$; en el caso de la referencia [3] los argumentos presentados van de 0.0 a 9.9 en intervalos de 0.1.

Sin embargo, en casos prácticos es necesario obtener valores fiables para las funciones $J_n(x)$ para cualquier argumento y órdenes que en algunos casos pueden ser de 20. (Por ejemplo en problemas de electro-dinámica con simetría axial).

Para ver los problemas de la pérdida de precisión empleando (1), consideremos un ejemplo.

Supongamos que deseamos calcular $J_8(2)$. Para el caso, buscamos $J_o(2)$ y $J_1(2)$ en la tabla de la referencia [3].

En efecto, $J_o(2) = 0.2239$, $J_1(2) = 0.5767$.

La recurrencia (1) proporciona $J_2(2)$ por medio de:

$J_2(2) = 2\times1/2\ J_1(2)-J_o(2) = 0.3528$, iterando:

$J_3(2) = 2\times2/2\ J_2(2)-J_1(2) = 0.1289$ ..

$J_4(2) = 2\times3/2\ J_3(2)-J_2(2) = 0.0339$

$J_5(2) = 2\times4/2\ J_4(2)-J_3(2) = 0.0067$

$J_6(2) = 2\times5/2\ J_5(2)-J_4(2) = -4\times10^{-4}$

$J_7(2) = 2\times6/2\ J_6(2)-J_5(2) = -0.0091$

$J_8(2) = 2\times7/2\ J_7(2)-J_6(2) = -0.0641.$

Buscando en tablas más completas [4], los valores para las funciones calculadas son:

$J_2(2)=0.3528$; $J_3(2)=0.1289$; $J_4(2)=0.03399$; $J_5(2)=0.0070$; $J_6(2)=0.0012$; $J_7(2)=0.000175$; $J_8(2)=0.0000222$.

Como puede apreciarse, a partir de la tercera iteración la precisión "flaquea" y en la quinta ya es un "desastre".

El haber seleccionado el argumento como $x = 2$ es simple comodidad, y la situación mostrada es totalmente general.

El problema reside en restar dos cantidades muy parecidas, resultando una propagación de errores tal, que el resultado carece de sentido.

Podría pensarse que al emplear un número mayor de dígitos el problema se subsanaría. De safortunadamente **ese no es el caso**, como el **lector** lo puede constatar **fácilmente**.

3. METODO NUMERICO.

Es un algoritmo **para el cálculo** de $J_n(x)$ tanto para cualquier **argumento**, (lo que evita engorrosas interpolaciones por medio de tablas), así como para todo orden, eludiendo además errores difíciles de controlar al utilizar la definición por series.

El método sugerido [4] consiste en lo siguiente:

Primero que todo hay que tener en cuenta que las funciones Bessel de primera clase satisfacen la siguiente relación:

$$J_n(0) = \begin{cases} 1, & n = 0 \\ 0, & n = 1,2,3,\ldots \end{cases} \tag{3}$$

En cuanto a la simetría respecto al orden se tiene que:

$$J_{-n}(x) = (-1)^n J_n(x) \tag{4}$$

y además cumplen la siguiente relación que puede ser interpretada como una "normalización":

$$J_o(x) + \sum_{k=1}^{\infty} 2J_{2k}(x) = 1 \qquad (5)$$

En este momento se introduce una función auxiliar $\tilde{J}$ que satisface la misma fórmula de recurrencia que las funciones de Bessel (1) con la característica adicional que, para un M (orden de la función auxiliar) lo suficientemente grande se tenga:

$$\tilde{J}_{M+1}(x) = 0; \quad \tilde{J}_M(x) = 1 \qquad (6)$$

independientemente del valor del argumento.

De (5) también es claro que $\lim_{n\to\infty} J_n(x) = 0$; por lo tanto dado x, para n suficientemente grande $J_n(x) \sim 0$. Además, como las funciones $\tilde{J}$ y J poseen la misma fórmula de recurrencia se tendrá que $\tilde{J}_n(x) \sim C\ J_n(x)$, en donde la constante C de proporcionalidad se obtiene de la "normalización" (5).

Dado que las funciones auxiliares satisfacen la recurrencia (1), con los valores de la expresión (6) es posible iterar en reversa hasta $\tilde{J}_o(x)$ por medio de :

$$\tilde{J}_{n-1}(x) = \frac{2n}{x} \tilde{J}_n(x) - \tilde{J}_{n+1}(x) \qquad (7)$$

Simultáneamente se va acumulando el valor de la ecuación (5), esta vez para los $\tilde{J}$, de tal suerte que:

$$S = \tilde{J}_o(x) + \sum_{k=1}^{M} 2\tilde{J}_{2k}(x) \tag{8}$$

Esta última relación permite conectar el valor $\tilde{J}_n$ con el valor de la función que se quiere calcular por medio de la simple conversión:

$$J_n(x) = S^{-1}\tilde{J}_n(x).$$

Con el presente método numérico, los resultados obtenidos *son exactos* hasta las *ocho primeras cifras decimales.*

La convergencia del método descrito anteriormente, es claramente sensible con la "definición del infinito" hecha por medio de M. Adicionalmente, si se ha de tener en cuenta simultáneamente rapidez y precisión, la escogencia de M juega un papel importante.

Una salida evidente, sería asignar a M un valor definitivamente grande, digamos 100, y así evitar problemas. Sin embargo, el tiempo de cálculo se incrementa apreciablemente, para la máquina para la cual se diseñó el programa que se reseña en el siguiente numeral, o para una similar.

Analizando la expresión para $J_n(x)$ dada en la ecuación (2), se nota que la convergencia de tal serie está gobernada por el valor del argumento.

La manera entonces, de seleccionar M, fué ensayando reiteradamente de acuerco con el rango del argumento qué valor de M lograba mantener la precisión mencionada. Esto se condensa en las líneas 100 a 120 del programa que a continuación se describe.

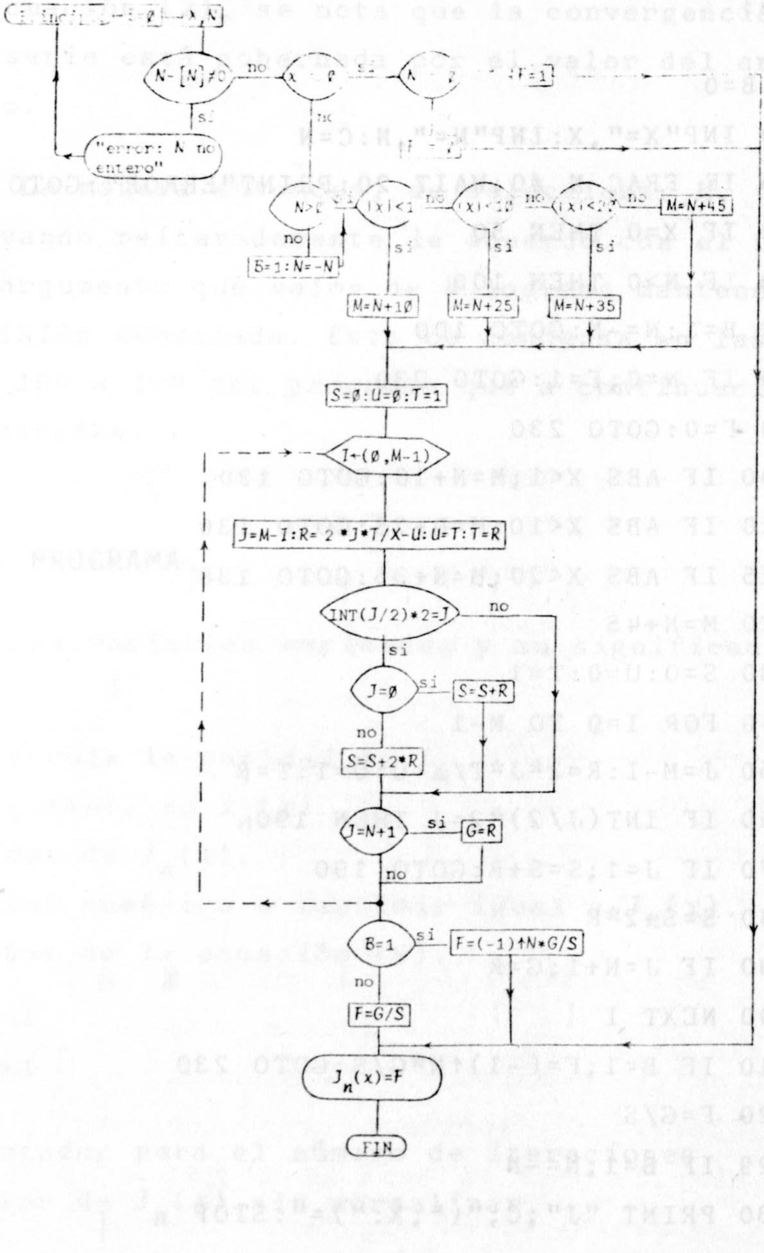
4. EL PROGRAMA.

Las variables empleadas y su significado son:

B: Controla la paridad de J_n.
X: Argumento en $J_n(x)$.
N: Orden de $J_n(x)$.
F: Valor numérico a imprimir igual a $J_n(x)$.
M,S: Las de la ecuación (8).
R: $\tilde{J}_{n-1}$
U: $\tilde{J}_{n+1}$
T: $\tilde{J}_n$
J: Contador para el número de iteraciones.
G: Valor de $\tilde{J}_n(x)$ sin normalizar.

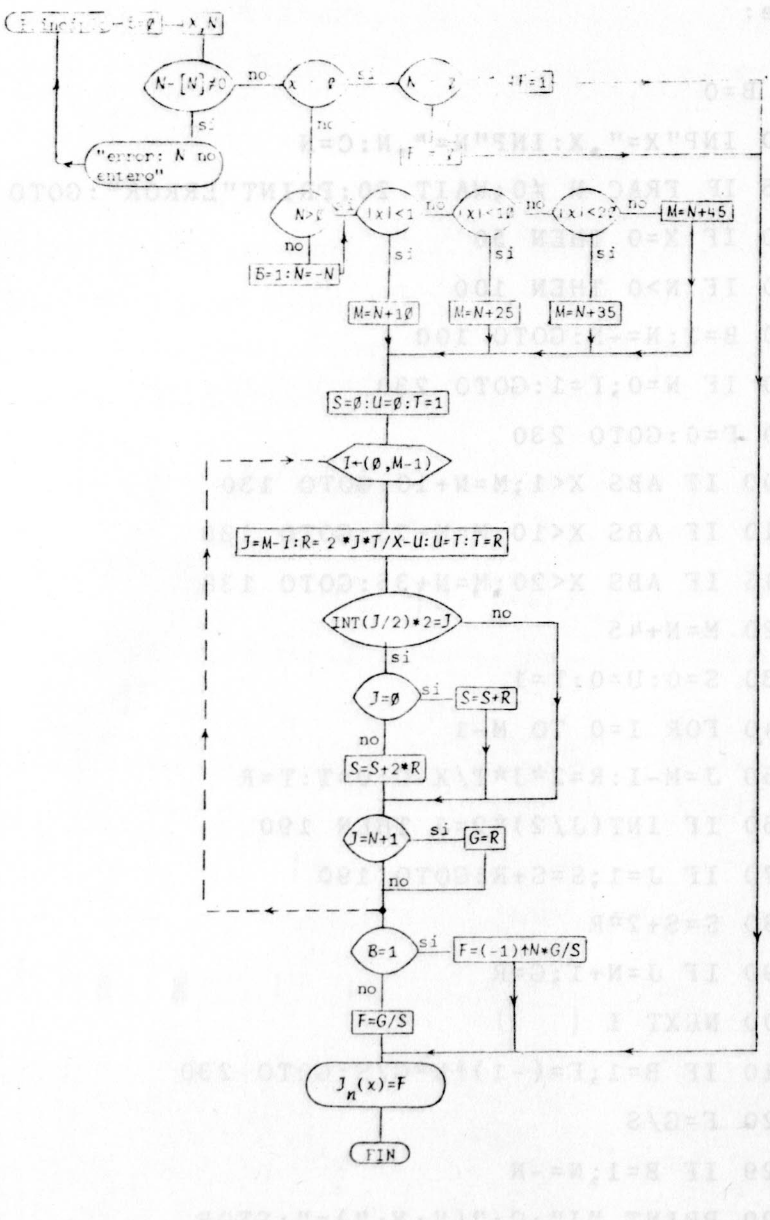
Un diagrama de flujo del programa se anexa al final del artículo. El listado es el siguiente:

```
5 B=0
10 INP"X=",X:INP"N=",N:C=N
15 IF FRAC N ≠0;WAIT 20:PRINT"ERROR":GOTO 10
20 IF X=0 THEN 50
30 IF N>0 THEN 100
40 B=1:N=-N:GOTO 100
50 IF N=0;F=1:GOTO 230
60 F=0:GOTO 230
100 IF ABS X<1;M=N+10:GOTO 130
110 IF ABS X<10;M=N+25:GOTO 130
115 IF ABS X<20;M=N+35:GOTO 138
120 M=N+45
130 S=0:U=0:T=1
140 FOR I=0 TO M-1
150 J=M-I:R=2*J*T/X-U:U=T:T=R
160 IF INT(J/2)*2=J THEN 190
170 IF J=1;S=S+R:GOTO 190
180 S=S+2*R
190 IF J=N+1;G=R
200 NEXT I
210 IF B=1;F=(-1)↑N*G/S:GOTO 230
220 F=G/S
229 IF B=1;N=-N
230 PRINT "J";C;"(";X:")=":STOP
231 PRINT F
```


N-[N]≠0
no
si
"error: N no entero"
nc
N>Ø
|x|<1
|x|<10
|x|<20
M=N+45
no
si
B=1:N=-N
M=N+1Ø
M=N+25
M=N+35
S=Ø:U=Ø:T=1
I←(Ø,M-1)
J=M-I:R=2*J*T/X-U:U=T:T=R
INT(J/2)*2=J
J=Ø
S=S+R
S=S+2*R
J=N+1
G=R
B=1
F=(-1)↑N*G/S
F=G/S
J_n(x)=F
FIN

al final del artículo. El listado es el siguiente:

```
5 B=0
10 INP"X=",X:INP"N=",N:C=N
15 IF FRAC N ≠0;WAIT 20:PRINT"ERROR":GOTO 10
20 IF X=0 THEN 50
30 IF N>0 THEN 100
40 B=1:N=-N:GOTO 100
50 IF N=0;F=1:GOTO 230
60 F=0:GOTO 230
100 IF ABS X<1;M=N+10:GOTO 130
110 IF ABS X<10;M=N+25:GOTO 130
115 IF ABS X<20;M=N+35:GOTO 138
120 M=N+45
130 S=0:U=0:T=1
140 FOR I=0 TO M-1
150 J=M-I:R=2*J*T/X-U:U=T:T=R
160 IF INT(J/2)*2=J THEN 190
170 IF J=1;S=S+R:GOTO 190
180 S=S+2*R
190 IF J=N+1;G=R
200 NEXT I
210 IF B=1;F=(-1)↑N*G/S:GOTO 230
220 F=G/S
229 IF B=1;N=-N
230 PRINT "J";C;"(";X:")=":STOP
231 PRINT F
```


N-[N]≠0
no
si
"error: N no entero"
no
si
F=1
N>1
|x|<1
|x|<10
|x|<20
M=N+45
B=1:N=-N
M=N+10
M=N+25
M=N+35
S=0:U=0:T=1
I+(0,M-1)
J=M-I:R= 2 *J*T/X-U:U=T:T=R
INT(J/2)*2=J
J=0
S=S+R
S=S+2*R
J=N+1
G=R
B=1
F=(-1)↑N*G/S
F=G/S
J_n(x)=F
FIN

REFERENCIAS

[1] Takeuchi, Y., Matemática enseñanza universitaria, Nº 34, p.47, (1985).

[2] García, M., Rev. Col. de Física, 11 p.103, (1977).

[3] Spiegel, M., *Mathematical Handbook*, McGraw Hill Book, (1968).

[4] Abramovitz-Stegun., *Handbook of Mathematical Functions*, Dover, (1972).

*

Departamento de Física
Universidad Nacional de Colombia

BOGOTA. D.E.