

EL LENGUAJE ALGORITMICO PUC-R2

por

CARLOS DOMINGO

1.- Lenguajes algorítmicos vs. lenguajes simbólicos. En el estado actual de la tecnología de las computadoras hay dos formas esenciales de codificar la solución de un problema:

a) Usando un lenguaje algorítmico y un compilador que traduce las macro instrucciones del programa en lenguaje algorítmico a instrucciones de máquina.

b) Usando un lenguaje simbólico y un ensamblador que traduce cada instrucción simbólica en la correspondiente instrucción de la máquina.

Resumamos en un cuadro las características ventajas e inconvenientes de cada uno de éstos métodos.

LENGUAJE ALGORITMICO

A una instrucción simbólica le corresponden muchas instrucciones de máquina.

Orientado hacia el problema.

Se aprende fácilmente.

Su sencillez hace que se cometan pocos errores de programación.

Programas fuente fácilmente inteligibles.

Similitud con las fórmulas algebraicas.

Los lenguajes para usar en diferentes máquinas son parecidos, y a veces casi iguales, lo cual facilita el aprendizaje y cambios de programas.

Los programas resultantes son, en general, poco eficientes desde el punto de vista de tiempo y memoria ocupados.

El programa objeto es difícil de interpretar. No se sabe qué procedimiento se usa para cumplir

LENGUAJE SIMBOLICO

A una instrucción simbólica le corresponde una instrucción de máquina.

Orientado hacia la máquina.

Es de aprendizaje dificultoso.

Los errores de programación son frecuentes.

Programas difíciles de entender.

No hay similitud evidente del programa con la expresión algebraica del problema.

Los lenguajes para diferentes máquinas son, en general, diferentes.

Se puede alcanzar la máxima eficiencia compatible con la máquina y la habilidad del programador.

Dominio completo del programa sobre el lenguaje objeto. Este no es más difícil de interpretar que el len-

las instrucciones de la máquina.

Debido a lo anterior, es casi imposible hacer parches en un Programa objeto.

Apto para el tipo de problemas para el cual fue diseñado, el lenguaje (problemas matemáticos o comerciales o de manejo de información). La entrada y salida de datos tiene un tipo de formato predeterminado.

lenguaje simbólico.

Las correcciones y parches al programa objeto se hacen fácilmente.

Apto para todo tipo de problemas manejables con computadoras, incluyendo construcción de compiladores y ensambladores.

El formato de entradas y salidas puede ser arbitrario.

De la lista anterior, que no es exhaustiva, se comprende que la elección de que tipo de lenguaje debe usarse en circunstancias dadas es un problema estratégico. En él intervienen variables como dimensión y duración posible del programa, número de veces que se usará, tiempo y personal disponible para la programación, tipo de problema, formato de datos y resultados, etc.

El problema es especialmente agudo en una Universidad donde debe prevverse la posibilidad de toda clase de problemas desde tabular hasta clasificar exámenes o analizar textos y donde pueden esperarse usuarios con los más diversos grados de preparación, y las exigencias más variadas, en una palabra donde el trabajo desborda toda rutina.

También ocurre con frecuencia que en un mismo problema hay partes en que sería deseable usar un lenguaje simbólico y partes en que el lenguaje algorítmico es más adecuado.

2.- Sistemas de programación con lenguajes de niveles mezclados.- Todas estas consideraciones han llevado a la idea de desarrollar sistemas de programación en que los dos niveles de lenguaje puedan mezclarse libremente. (1)

En nuestro departamento de Cálculo se ha hecho un primer ensayo de desarrollar un sistema de este tipo para la IBM-1620. En un año de trabajo se ha logrado implementar el PUC-R2 cuyas características esenciales son las siguientes:

- 1- Al escribir un programa en PUC-R2 se permite mezclar libremente instrucciones simbólicas (tipo SPS) con macro-instrucciones (tipo ALGOL con ciertas restricciones).
- 2- El procesador traduce todas las instrucciones a lenguaje simbólico.
- 3- El programa en lenguaje simbólico se traduce a lenguaje de máquina en la forma usual.

Los criterios que se han seguido en este sistema son:

- a) Permitir al programador un control y comprensión completos del programa ensamblado. Para este fin se traduce todo el programa a lenguaje

simbólico y el programa traducido es análogo al programa tal como lo realiza ría un programador. Esto facilita las correcciones y modificaciones.

- b) Permite el uso de instrucciones o macro-instrucciones de acuerdo con la conveniencia del programa. Por ejemplo, una expresión aritmética que se calcula unas pocas veces en el programa, se programará como una macro-instrucción del lenguaje algorítmico. Una expresión que está dentro de un lazo iterativo (o de varios lazos) se podrá programar en lenguaje simbólico buscando la máxima eficiencia.
- c) Permitir el uso de todas las posibilidades de la máquina.
- d) Producir un programa objeto de la mayor eficiencia posible.

Creemos que estos objetos han sido alcanzados y es pertinente preguntar a qué precio. La respuesta es inmediata:

- 1- El PUC-R2 es un compilador de tres pasos
- 2- El PUC-R2 requiere 60 K de memoria.

Sin embargo, no creemos que ambas características sean inevitables y de hecho la primera está superada en la versión 3.

3.- Estructura del PUC-R2. Daremos los tipos de instrucciones con algunos ejemplos:

- a) Instrucciones simbólicas. Son análogas a las del lenguaje SPS-1 de IBM pero pueden usarse en ellas variables con subíndices.

Ejemplos:

```
TF ALFA+10,BETA,11
AM ALFA,72150
FD B,K(I,M1)
```

- b) Macro-instrucciones simbólicas.

- Declaraciones

```
REAL DECL 10R,10B,10I
INTGER DECL 5J,2L,2Z
REAL ARRY 10BETA(5,23),10A(12)
```

Toda variable que se utilice en las macro-instrucciones deberá ser previamente declarada.

-Instrucciones aritméticas.

```
XXL ARIT A1=R+Z*BETA(I,J)**L
ARIT Z=SIN(X**2-ABS(COS(A(J))))+1.0
```

-Instrucciones condicionales. Pueden ser designativas de un rótulo o

asignativas (dan un valor a una variable)

```
IF AG LSS B+50 THEN LABEL1 ELSE W1
W1 IF A(K)+R GTR SIN(B) THEN (Z=L+J)
IF ALFA LEQ 1.0 THEN (ALFA=1) ELSE (A=0)
```

En el primer ejemplo si $AG < B+50$ se ramifica a la instrucción de dirección simbólica LABEL1, en caso contrario se ramifica a W1.

En el segundo ejemplo si $A(K)+R > SIN(B)$, se hace $Z=L+J$; en caso contrario se pasa a la próxima instrucción.

En el tercer ejemplo, se $ALFA \leq 1.0$ se hace $ALFA = 1$; en caso contrario se hace $A = 0$; luego se pasa a la próxima instrucción.

-Instrucciones iterativas. Hay tres tipos:

```
LOOP1 FOR X=1.0 STEP H UNTIL Z=0.5 DO Y2
Y2 ARIT H=H/2
FOR X1=X1/2 WHILE X1 GTR 0.001 Z DO ALFA
ALFA
FOR Z=A+B,2,0,Z-1.0 DO FINAL
FINAL
```

En el primer ejemplo se ejecutan todas las instrucciones del rango del FOR (hasta el rótulo Y2 inclusive) con valores $X=1.0$; $X=1.0+H$; $X=1.0+H+H, \dots$ hasta que $Z > X-0.5$.

En el segundo ejemplo, se ejecutan todas las instrucciones del rango del FOR, haciendo el principio de cada ciclo $X1=X1/2$, y la repetición se continúa mientras $X1 > 0.001 Z$.

En el tercer ejemplo, se ejecutan todas las instrucciones del rango, haciendo la primera vez $Z=A+B$, la segunda $Z=0$, la tercera $Z=Z-1.0$.

-Llaves.

```
S1 SWCHB=FINAL,XX1,W1,LABEL
```

(Se bifurca a FINAL, XX1, W1, LABEL según que el valor actual de B sea 1, 2, 3, ó 4 respectivamente)

-Instrucciones de entrada y salida.

El uso de instrucciones básicas permite la máxima diversidad y eficiencia en salida y entrada de datos. Sin embargo, para comodidad del programador se dispone de dos macro-instrucciones de entrada y salida.

TIT1 READ A, ENTERO, Q(I, J), BETA, (M, N, P, Q)

En este ejemplo se leen las siguientes variables que están en formato libre A, ENTERO, el elemento de arreglo Q(I, J) y el arreglo BETA, con los subíndices variando de M a N y de P a Q. En el formato libre los datos deben estar ignorados por uno o más blancos. Los datos alfabéticos son ignorados.

TIT2 PNCH FORM4, L, Q(I, J)

Se perforan los valores L y Q(I, J) de acuerdo con un formato expresado por la constante alfabética FORM4, que es una máscara de edición de 80 columnas. Por ejemplo:

```
FORM4 DAC 40,bbb0000bbb00.00000
DAC 40,
```

4.- Ejemplo de programa en PUC-R2. El ejemplo siguiente es un programa para calcular las funciones de BESSEL de órdenes M, R, R+1 (enteros positivos) entre valores de X dados (X_I y X_F) y con la precisión prefijada EPSI. El programa se basa en la serie de términos alternados:

$$J_N(X) = \sum_{S=0}^{\infty} \frac{(-1)^S}{S!(N+S)!} \left(\frac{X}{2}\right)^{N+2S}$$

procede partiendo del término (S=0)

$$\frac{1}{N!} \left(\frac{X}{2}\right)^N$$

y multiplicando, para hallar cada término sucesivo, el anterior por

$$\frac{1}{(S+1)(S+1+N)} \left(\frac{X}{2}\right)^2.$$

Son de notar varias particularidades que en la programación se manejan en forma muy clara.

- 1- El cálculo de N! no se hace cuando N=0 y N=1, pues entonces es N!=1. Esto se hace automáticamente, ya que el proceso del FOR si el valor que adopta la variable de control es superior al valor máximo, las instrucciones del rango del FOR no se realizan.
- 2- En el caso N=0 y X=0, el cálculo por series conduce a la expresión indeterminada 0⁰. Sin embargo J₀(0)=1; esta posibilidad se considera en la decisión lógica:

```
IF ABS(X)+N NEQ 0 THEN GRØXNZ
```

que salta la igualación J₀(0)=1.0 en el caso en que ó bien X o bien N ó ambos sean diferentes de cero.

3- El primer término $TERM = (X/2)^N/N!$ produce error cuando $X=0$ en el cálculo usual por medio de logaritmos. En este caso, el uso del IF asignativo es obvio.

4- Obsérvese la libre mezcla de modos que hacen el programa claro (a costa, por supuesto, de las transformaciones de modo necesarias).

Veáse el desarrollo del programa en las páginas 19-25.

5.- Algunos ejemplos sobre optimización. Damos algunos ejemplos en los cuales el uso del lenguaje simbólico básico es fácil y produce notable optimización.

a) Expresiones iterativas. Es a veces conveniente programarlas en lenguaje simbólico:

```
TIT4   FOR   N=R STEP-2 UNTIL O DØ FINI
      . . . . .
FINI   . . . . .
```

Puede programarse

```
TIT4   TF   N,R
      . . . . .
FINI   . . . . .
      SM   N,2
      BNN  TIT4+12
```

b) Expresiones con subíndices. Se puede evitar el cálculo del mismo subíndice.

```
ARIT A(I,J)=A(I,J)+BR
```

que da lugar a los cálculos de índices una suma y una transferencia puede programarse en la siguiente instrucción:

```
FA     A(I,J),BR
```

que solo da lugar a un cálculo de índices y una suma.

c) La instrucción aritmética en modo entero.

```
ARIT N=N+1
```

que da origen a tres instrucciones simbólicas y al almacenamiento de una constante (en FORTRAN II da lugar a unas 14 instrucciones y una constante), se programaría en la única instrucción:

```
AM     N,1
```

d) La instrucción condicional

```
IF     A GTR B THEN TIT8
```

que da origen a 4 instrucciones en PUC-R2 se programa facilmente en las dos instrucciones:

A,B

BH TITB

Observese que en este caso la eficiencia del FORTRAN II no es muy grande. La instrucción:

IF(A-B)6,6,8

que cumpliría las funciones de la anterior implica una resta en punto flotante lo cual son 25 ó 30 instrucciones.

6.- Comparación con el Fortran II.-

Se han realizado varios programas en FORTRAN II y en PUC-R2 con objeto de comparar el comportamiento del programa objeto. En el siguiente cuadro se muestran los resultados obtenidos. Excepto en el programa de cálculo de pórticos no se ha procurado optimizar el programa en PUC-R2 mediante uso de lenguaje básico.

<u>PROGRAMA</u>	<u>M E M O R I A</u>		<u>TIEMPO DE UNA CORRIDA</u>	
	<u>FORTRAN II</u>	<u>PUC-R2</u>	<u>FORTRAN II</u>	<u>PUC-R2</u>
Solución sistemas Lineales (CROUT)	18612 error 0.01	12030 error 0.001	10 Seg.	6 seg.
Exploración de plano complejo (Problema de Física)	14200	6280	20 min.	10 min.
Cálculo de Pórticos Método del Profesor Fortul.	37000	28000	4 min.30 seg.	2 min. 30 seg.
Cálculo de frente óptico de explotación en minas	22426	17400	9 min.	5min. 15 seg.

El ahorro de tiempo y memoria depende naturalmente de la destreza del programador, del tiempo de que dispone para programar y de la naturaleza del problema.

PROGRAMA DEL EJEMPLO DEL NUMERAL 4

```

01 010 *
01 020 * EL SIGUIENTE PROGRAMA TABULA LAS FUNCIONES DE BESSEL DE
01 030 * ORDENES M,R,R+1 SIENDO M,R ENTEROS POSITIVOS.
01 031 * EL CALCULO SE HACE ENTRE LOS VALORES XI,XF.
01 040 * EL ERROR ES MENOR QUE EPSI.
01 050 *
01 060 * DECLARACION DE VARIABLES
01 070 *
01 080 INTGER DECL 3M,3N,3R,10NFAC,3I,3S
01 090 REAL DECL 10J,10TERM,10X,10XI,10XF,10G,10EPSI,10DELTX
01 100 FORM DAC 40,N= 000 X= 00.00000 J= 0.000000 Z
01 110 DAC 41, -
01 120 *
01 130 * LECTURA DE DATOS
01 140 *
01 150 LECT READ M,R,XI,XF,DELTX,EPSI
02 010 *
02 020 * CALCULO DE J
02 030 *
02 040 FOR N=M,R,R+1 DO PERF
02 050 FOR X=XI STEP DELTX UNTIL XF DO PERF
02 060 ARIT NFAC=1
02 070 FOR I=2 STEP 1 UNTIL N DO CALFAC CALCULO FACTORIAL
02 080 CALFAC ARIT NFAC=NFAC*I (SE SALTA SI N=0,1)
02 090 IF ABS(X)+N NEQ 0. THEN GROXNZ VER SI GRADO Y ARG. SON CERO
02 100 ARIT J=1.0 SI SON ES J0(0)=1
02 110 B PERF,,, IR A PERFORARLO
02 120 DORG *-3
02 140 GROXNZ ARIT G=(X/2.0)**2
02 150 IF X EQL 0.0 THEN (TERM=0.0) ELSE (TERM=(X/2.0)**N/NFAC)
02 155 ARIT J=TERM
02 160 ARIT S=0
02 170 FOR TERM=-TERM*/((S+1)*(S+1+N)) WHILE ABS(TERM) GTR EPSI DO JJ
02 180 ARIT S=S+1
02 190 JJ ARIT J=J+TERM
02 200 PERF PNCH FORM,N,X,J
02 210 H
02 220 B LECT
02 230 DEND LECT

```

```

01 010 *
01 020 *           EL SIGUIENTE PROGRAMA TABULA LAS FUNCIONES DE BESSEL DE
01 030 *           ORDENES M,R,R+1 SIENDO M,R ENTEROS POSITIVOS.
01 040 *           EL CALCULO SE HACE ENTRE LOS VALORES XI,XF.
01 050 *           EL ERROR ES MENOR QUE EPSI.
01 060 *
01 070 *           DECLARACION DE VARIABLES
01 080 *
01 090 *NTGER  DECL  3M,3N,3R,10NFAC,3I,3S

01 10G M      DS   3
01 11G N      DS   3
01 12G R      DS   3
01 13G NFAC   DS  10
01 14G I      DS   3
01 15G S      DS   3

01 160 *EAL   DECL  10J,10TERM,10X,10XI,10XF,10G,10EPSI,10DELTX

01 17G J      DS  10
01 18G TERM   DS  10
01 19G X      DS  10
01 20G XI     DS  10
01 21G XF     DS  10
01 22G G      DS  10
01 23G EPSI   DS  10
01 24G DELTX  DS  10

01 250 FORM   DAC  40,N= 000      X= 00.00000      J= 0.000000  Z
01 260        DAC  41,          -Z
01 270 *
01 280 *           LECTURA DE DATOS
01 290 *
01 300 *ECT   READ  M,R,XI,XF,DELTX,EPSI

01 31G LECT   TFM  READ1.,**+23
01 32G        B    READ.,-M
01 33G        DSA  3
01 34G        DC   1,-
01 35G        TFM  READ1.,**+23
01 36G        B    READ.,-R
01 37G        DSA  3
01 38G        DC   1,-
01 39G        TFM  READ1.,**+23
01 40G        B    READ.,XI
01 41G        DSA  10
01 42G        DC   1,-
01 43G        TFM  READ1.,**+23
01 44G        B    READ.,XF
01 45G        DSA  10
01 46G        DC   1,-
01 47G        TFM  READ1.,**+23
01 48G        B    READ.,DELTX
01 49G        DSA  10
01 50G        DC   1,-
01 51G        TFM  READ1.,**+23
01 52G        B    READ.,EPSI

```

```

01 53G DSA 10
01 54G DC 1,-

01 550 *
01 560 *          CALCULO DE J
01 570 *
01 580 *          FOR N=M,R,R+1 DO PERF

01 59G TFM F.R002+6,F.R002+8
01 60G TF N,ZE,-17
01 61G A N,M

01 620 *.R001 FOR X=XI STEP DELTX UNTIL XF DO PERF

01 63G F.R001 TF X,XI
01 64G F.R005 TF TEM.01,ZE.-10
01 65G C TEM.01,DELT X
01 66G BE **48
01 67G TF TEM.01,UNF.
01 68G BNH **24
01 69G SF TEM.01
01 70G TF TEM.02,X
01 71G FS TEM.02,XF
01 72G FM TEM.01,TEM.02
01 73G TF IF.A,99
01 74G TF IF.B,ZE.-10
01 75G C IF.A,IF.B
01 76G BH F.R006

01 770 *          ARIT NFAC=1

01 78G TF NFAC,ZE.-10
01 79G A NFAC,LI.001

01 800 *          FOR I=2 STEP 1 UNTIL N DO CALFAC          CALCULO FACTORIAL

01 81G TF I,ZE.-17
01 82G A I,LI.002
01 83G F.R008 TF TEM.01,ZE.-18
01 84G C TEM.01,LI.001
01 85G BE **48
01 86G TDM TEM.01,1
01 87G BNH **24
01 88G SF TEM.01
01 89G TF TEM.02,I
01 90G S TEM.02,N
01 91G M TEM.01,TEM.02
01 92G TF TEM.02,99
01 93G TF IF.A,ZE.-00
01 94G A IF.A,TEM.02
01 95G TF IF.B,ZE.-00
01 96G A IF.B,ZE.-18
01 97G C IF.A,IF.B
01 98G BH F.R009

```

```

01 990 *ALFAC ARIT NFAC=NFAC*I (SE SALTA SI N=0,1)
02 00G CALFAC M NFAC,I
02 01G TF TEM,01,99
02 02G TF NFAC,ZE,-10
02 03G A NFAC,TEM,01
02 04G TF TEM,01,I
02 05G A TEM,01,LI,001
02 06G TF I,ZE,-17
02 07G A I,TEM,01
02 08G B F,R008
02 09G DORG *-3
02 10G F.R009 NOP
02 11G DORG *-11

02 120 * IF ABS(X)+N NEQ 0. THEN GROXNZ VER SI GRADO Y ARG. SON CERO
02 13G TF TEM,01,X
02 14G CF TEM,01
02 15G TFM CAM1.0+35,**+23
02 16G B CAM1.0,N,7
02 17G FA TEM,01,39
02 18G TF IF,A,TEM,01
02 19G TF IF,B,ZE,-10
02 20G C IF,A,IF,B
02 21G BNE GROXNZ

02 220 * ARIT J=1.0 SI SON ES J0(0)=1
02 23G TF J,UNF.

02 240 B PERF,,, IR A PERFORARLO
02 250 DORG *-3
02 260 *ROXNZ ARIT G=(X/2.0)**2

02 27G GROXNZ FD X,LI,003
02 28G TF TEM,01,99
02 29G FM TEM,01,TEM,01
02 30G TF G,99

02 310 * IF X EQL 0.0 THEN (TERM=0.0) ELSE (TERM=(X/2.0)**N/NFAC)
02 32G TF IF,A,X
02 33G TF IF,B,ZE,-10
02 34G C IF,A,IF,B
02 35G BNE IF,004
02 36G TF TERM,ZE,-10
02 37G B IF,005
02 38G DORG *-3
02 39G IF.004 FD X,LI,003
02 40G TF TEM,01,99
02 41G TFM CAM1.0+35,**+23
02 42G B CAM1.0,N,7
02 43G FLOG TEM,01,TEM,01

```

```

02 44G      FM      TEM.01,39
02 45G      FEXT    TEM.01,99
02 46G      TFM     CAM1.0+35,**+23
02 47G      B       CAM1.0,NFAC,7
02 48G      FD      TEM.01,39
02 49G      TF      TERM,99
02 50G  IF.005  NOP
02 51G      DORG    *-11

02 520  *      ARIT  J=TERM

02 53G      TF      J,TERM

02 540  *      ARIT  S=0

02 55G      TF      S,ZE,-17
02 56G      A       S,ZE,-18

02 570  *      FOR   TERM=-TERM*G/((S+1)*(S+1+N)) WHILE ABS(TERM) GTR EPSI DO JJ

02 58G  F.R011  FM      TEM,G
02 59G      TF      TEM.01,99
02 60G      TF      TEM.02,S
02 61G      A       TEM.02,LI.001
02 62G      TF      TEM.03,S
02 63G      A       TEM.03,LI.001
02 64G      A       TEM.03,N
02 65G      M       TEM.02,TEM.03
02 66G      TF      TEM.03,99
02 67G      TFM     CAM1.0+35,**+23
02 68G      B       CAM1.0,TEM.03,7
02 69G      FD      TEM.01,39
02 70G      TF      TEM.01,99
02 71G      TF      TEM.03,ZE,-10
02 72G      S       TEM.03,TEM.01
02 73G      TF      TERM,TEM.03
02 74G      TF      TEM.01,TERM
02 75G      CF      TEM.01
02 76G      TF      IF.A,TEM.01
02 77G      TF      IF.B,EPSI
02 78G      C       IF.A,IF.B
02 79G      BNH    F.R012

02 800  *      ARIT  S=S+1

02 81G      TF      TEM.01,S
02 82G      A       TEM.01,LI.001
02 83G      TF      S,ZE,-17
02 84G      A       S,TEM.01

02 850  *J     ARIT  J=J+TERM

02 86G  JJ     TF      TEM.01,J
02 87G      FA     TEM.01,TERM

```

02	88G		TF	J,TEM.01
02	89G		B	F.R011
02	90G		DORG	*-3
02	91G	F.R012	NOP	
02	92G		DORG	*-11
02	930	*ERF	PNCH	FORM,N,X,J
02	94G	PERF	TFM	P.NCHX+04,FORM
02	95G		TFM	P.NCHX+09,N
02	96G		TFM	P.NCHX+14,X
02	97G		TFM	P.NCHX+19,J
02	98G		TFM	P.NCH1,*+20
02	99G		B	P.NCH
03	00G		DORG	*-3
03	01G		TF	TEM.01,X
03	02G		FA	TEM.01,DELTX
03	03G		TF	X,TEM.01
03	04G		B	F.R005
03	05G		DORG	*-3
03	06G	F.R006	NOP	
03	07G		DORG	*-11
03	08G	F.R002	B	
03	09G		DORG	*-3
03	10G		TFM	F.R002+6,F.R003
03	11G		TF	N,ZE.-17
03	12G		A	N,R
03	13G		B	F.R001
03	14G		DORG	*-3
03	15G	F.R003	TFM	F.R002+6,F.R004
03	16G		TF	TEM.01,R
03	17G		A	TEM.01,LI.001
03	18G		TF	N,ZE.-17
03	19G		A	N,TEM.01
03	20G		B	F.R001
03	21G		DORG	*-3
03	220	F.R004	H	
03	230		B	LECT

D A T O S

N= 0 M= 1 X INICIAL= 0 X FINAL= 2 INCREMENTO= 0.2 EPSI= 0.000001

R E S U L T A D O S

N= 0	X= .00000	J= 1.000000
N= 0	X= .20000	J= .990025
N= 0	X= .40000	J= .960398
N= 0	X= .60000	J= .912004
N= 0	X= .80000	J= .846287
N= 0	X= 1.00000	J= .765197
N= 0	X= 1.20000	J= .671133
N= 0	X= 1.40000	J= .566855
N= 0	X= 1.60000	J= .455402
N= 0	X= 1.80000	J= .339985
N= 0	X= 2.00000	J= .223890
N= 1	X= .00000	J= .000000
N= 1	X= .20000	J= .099500
N= 1	X= .40000	J= .196026
N= 1	X= .60000	J= .286701
N= 1	X= .80000	J= .368841
N= 1	X= 1.00000	J= .440049
N= 1	X= 1.20000	J= .498289
N= 1	X= 1.40000	J= .541947
N= 1	X= 1.60000	J= .569896
N= 1	X= 1.80000	J= .581516
N= 1	X= 2.00000	J= .576724
N= 2	X= .00000	J= .000000
N= 2	X= .20000	J= .004983
N= 2	X= .40000	J= .019734
N= 2	X= .60000	J= .043665
N= 2	X= .80000	J= .075818
N= 2	X= 1.00000	J= .114903
N= 2	X= 1.20000	J= .159348
N= 2	X= 1.40000	J= .207355
N= 2	X= 1.60000	J= .256967
N= 2	X= 1.80000	J= .306144
N= 2	X= 2.00000	J= .352834

BIBLIOGRAFIA

- [1] EVANS Jr. - PERLIS, A. J.: The use of Threaded Lists in constructing a Combined ALGOL and Machine-Like Assembly Processor, A Communications of ACM, January, 1960.

(Recibido Octubre 1964)

UNIVERSIDAD CENTRAL DE
VENEZUELA, CARACAS.