Notes on the back story of this document

This set of **27** programs for the ***HP-34C*** was developed by *Eugenio E. Úbeda Forte* in September, 1980. They were sent to me by *Fernando del Rey*, who is as clueless about who this person is as I am. Since the author lived in Cartagena (Spain) at the time, F.'s guess is that probably he was one of his father's colleagues who (upon getting to know that F. liked the *HP-34C*) sent him the complete set so that he could have a look at it.

Whatever the details of its origin, the truly *excellent* typewritten documentation for the programs is both very extensive (123 pages, average 4.5 pages per program) and comprehensive, including for each program the sections *Objeto* (the program's purpose), *Método* (relevant equations and/or algorithms), *Observaciones* (comments and/or notes), *Ejemplos* (examples), *Instrucciones de uso* (usage instructions) and the full program listing, detailing registers, labels and flags used, plus angular and display settings. A categorized index of all programs featured is also included.

Whether this set has been published before, as a book or otherwise, is uncertain but very unlikely, so this is probably the first time it sees the light. The titles for all 5 categories and 27 programs follow, in both the original ***Spanish*** and in ***English***.

> ***Caveat lector****: All the documentation is in Spanish. However, as the subject matters are mostly mathematical in nature, it's perfectly possible to follow and run the examples with little difficulty, thus also learning how to use the programs.*

27 Programas para la HP-34C *(by Eugenio E. Úbeda Forte, Sept. 1980; 27 programs, 123 pages)*

1. Matrices y sistemas lineales.
Matrices and linear systems.

1.1. Multiplicación de una matriz cuadrada por su transpuesta.
Multiplication of a square matrix by its transpose matrix.

1.2. Factorización y determinante de una matriz cuadrada.
Square matrix factorization and determinant.

1.3. Factorización y determinante de una matriz simetrica.
Symmetric matrix factorization and determinant.

1.4. Solución de dos ecuaciones simultáneas con coeficientes complejos.
Solving 2x2 linear systems with complex coefficients.

2. Polinomios.
Polynomials.

2.1. Suma de productos de orden N de M números reales.
Sum of N^{th}-order products of M real numbers.

2.2. *Algoritmo de Horner.*
Horner's method.

2.3. *Algoritmo de Horner de doble fila.*
Two-row Horner's method.

2.4. *Recíproco de un desarrollo polinómico.*
Coefficients of the reciprocal of a polynomial.

2.5. *Potenciación de un desarrollo polinómico.*
Coefficients of the N^{th} power of a polynomial.

2.6. *Exponenciación de un desarrollo polinómico.*
Coefficients of the exponential of a polynomial.

2.7. *Estabilidad de polinomios: Criterio de Routh.*
Polynomial stability: Routh's Criterium.

2.8. *Evaluación de polinomios de Chebyshev y sus raices.*
Chebyshev's Polynomials evaluation and roots.

2.9. *Transformación de un polinomio en una expansión de Chebyshev.*
Transformation of a polynomial into a Chebyshev's expansion.

2.10. *Transformación de una expansión de Chebyshev en un polinomio.*
Transformation of a Chebyshev's expansion into a polynomial.

2.11. *Cambio de variable en polinomios.*
Change of variable in polynomials.

3. Interpolación y aproximación.
Interpolation and approximation.

3.1. *Interpolación polinómica.*
Polynomial interpolation.

3.2. *Desarrollo de Fourier de funciones con valores discretos*
Fourier Series expansion of a function given as a set of discrete data points.

3.3. *Desarrollo de Fourier de funciones discretizadas.*
Fourier Series expansion of a user-specified function f(x).

3.4. *Desarrollo de Fourier por integración de coeficientes.*
Fourier Series expansion coefficients obtained by integration.

3.5. *Desarrollo de Chebyshev de funciones discretizadas.*

Chebyshev Series expansion of a user-specified function f(x).

3.6. Desarrollo de Chebyshev por integración de coeficientes.
Chebyshev Series expansion coefficients obtained by integration.

4. Ecuaciones diferenciales.
Differential equations.

4.1. Solución de ecuaciones diferenciales de 1º orden.
Numeric solution of first-order differential equations.

4.2. Solución de sistemas de dos ecuaciones diferenciales de 1º orden.
Numeric solution of a system of two first-order differential equations.

4.3. Solución de ecuaciones diferenciales de 3º orden.
Numeric solution of third-order differential equations.

5. Varios.
Miscellaneous.

5.1. Filtros activos Sallen-Key de 2º orden.
Sallen-Key active filters of the second order.

5.2. Filtros activos Sallen-Key de 3º orden.
Sallen-Key active filters of the third order.

5.3. Equilibrado dinámico en dos planos.
Dynamic balancing on two planes.

Valentin Albillo, 23-10-2021

27 PROGRAMAS PARA LA HP - 34 C

Desarrollados y enviados por
Eugenio E. Ubeda Forte.-

Septiembre, 1980
CARTAGENA

INDICE

1.- Matrices y sistemas lineales.

1.1 - Multiplicación de una matriz cuadrada por su transpuesta.

1.2 - Factorización y determinante de una matriz cuadrada.

1.3 - Factorización y determinante de una matriz simétrica.

1.4 - Solución de dos ecuaciones simultáneas con coeficientes - complejos.

2.- Polinomios.

2.1 - Suma de productos de orden N de M números reales.

2.2 - Algoritmo de Horner.

2.3 - Algoritmo de Horner de doble fila.

2.4 - Recíproco de un desarrollo polinómico.

2.5 - Potenciación de un desarrollo polinómico.

2.6 - Exponenciación de un desarrollo polinómico.

2.7 - Estabilidad de polinomios: Criterio de Routh.

2.8 - Evaluación de polinomios de Chebyshev y sus raíces.

2.9 - Transformación de un polinomio en una expansión de Chebyshev.

2.10 - Transformación de una expansión de Chebyshev en un polinomio.

2.11 - Cambio de variable en polinomios.

3.- Interpolación y aproximación.

3.1 - Interpolación polinómica.

3.2 - Desarrollo de Fourier de funciones con valores discretos.

3.3 - Desarrollo de Fourier de funciones discretizadas.

3.4 - Desarrollo de Fourier por integración de coeficientes.

3.5 - Desarrollo de Chebyshev de funciones discretizadas.

3.6 - Desarrollo de Chebyshev por integración de coeficientes.

4.- Ecuaciones diferenciales.

4.1 - Solución de ecuaciones diferenciales de 1º orden.

4.2 - Solución de sistemas de dos ecuaciones diferenciales de 1º orden.

4.3 - Solución de ecuaciones diferenciales de 3º orden.

5.- Varios.

5.1 - Filtros activos Sallen-Key de 2º orden.

5.2 - Filtros activos Sallen-Key de 3º orden.

5.3 - Equilibrado dinámico en dos planos.

MULTIPLICACION DE UNA MATRIZ CUADRADA POR SU TRANSPUESTA.

1.- Objeto.

Dada una matriz cuadrada A, NxN, con $N \leq 4$, el siguiente programa obtiene, elemento por elemento, la matriz simétrica resultante de multiplicar A por su transpuesta A\`, bien por la izquierda (A\`A), bien por la derecha (AA\`).

La matriz NxN resultante es, como se ha dicho, simétrica y con la diagonal principal frecuentemente dominante, por lo que esta operación, entre otras cosas, se puede emplear para transformar sistemas lineales con objeto de asegurar la convergencia iterativa de su solución.

2.- Método.

Sea la matriz $A=(a_{ij})$ y su transpuesta $A^{`}=(a_{ji})$.

Si la multiplicación es por la izquierda, obtenemos:

$$B=A^{`}A = (b_{ij}) \quad , \quad b_{ij}= b_{ji} \quad , \quad b_{ij} = \sum_{k=1}^{N} a_{ki}\, a_{kj}$$

Si la multiplicación es por la derecha, obtenemos:

$$B = AA^{`} = (b_{ij}) \quad , \quad b_{ij}=b_{ji} \quad , \quad b_{ij} = \sum_{k=1}^{N} a_{ik}\, a_{jk}$$

3.- Observaciones.

Ya que la diferencia entre los dos productos es exclusivamente el orden al tomar los subíndices (ki, kj o ik, jk) bastará emplear un "flag" (el 0) para identificar el producto y tomar el orden adecuado.

La subrutina B se emplea para el direccionamiento de los elementos a_{ij}. La A, para su almacenamiento. Ante la manifiesta imposibilidad de almacenar los b_{ij}, éstos se van mostrando en el orden en que se calculan.

4.- Ejemplo.

Sea la matriz 4x4:

$$A= \begin{bmatrix} 5 & 0 & 2 & 3 \\ 2 & 4 & 1 & 1 \\ 1 & 3 & 10 & 2 \\ 0 & 7 & 5 & 6 \end{bmatrix}$$

Siguiendo las instrucciones, los elementos se almacenarán:

5; ENTER↑ ; 1; ENTER↑ ; 1; A.

0; ENTER↑ ; 2; ENTER↑ ; 1; A.

2; ENTER↑ ; 3; ENTER↑ ; 1; A.

3; ENTER↑ ; 4; ENTER↑ ; 1; A.

2; ENTER↑ ; 1; ENTER↑ ; 2; A.

4; ENTER↑ ; 2; ENTER↑ ; A.

1; ENTER↑ ; 3; ENTER↑ ; 2; A.

Etc.

Para la multiplicación por la izquierda haremos: 4; GSB 1.

Y se obtienen los sucesivos elementos de la matriz:

$$B = A`A = \begin{bmatrix} 30 & 11 & 22 & 19 \\ & 74 & 69 & 52 \\ & & 130 & 57 \\ & & & 50 \end{bmatrix}$$

indicando la flecha la simetría de los elementos restantes.

Para la multiplicación por la derecha haremos: 4; GSB 2.

Y obtenemos:

$$B = AA` = \begin{bmatrix} 38 & 15 & 31 & 28 \\ & 22 & 26 & 39 \\ & & 114 & 83 \\ & & & 110 \end{bmatrix}$$

El tiempo de ejecución para cada elemento es de unos 20 s.

HP-34C

TITULO: MULTIPLICACION DE UNA MATRIZ CUADRADA POR SU TRANSPUESTA.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR ELEMENTOS DE LA MATRIZ HACIENDO, PARA $i=1,\ldots,N$; $j=1,\ldots,N$ ($N \leq 4$)	a_{ij}	ENTER ↑	
		j	ENTER ↑	
		i	A	a_{ij}
3	INGRESAR ORDEN	N		
4.1	SI ES MULTIPLICACIÓN POR LA IZQUIERDA ($A'A$), HACER		GSB 1	
4.2	SI ES MULTIPLICACIÓN POR LA DERECHA (AA') HACER		GSB 2	
5	BREVEMENTE MOSTRARÁ MEDIANTE PAUSA			"i,j"
	Y DESPUÉS EL ELEMENTO DE LA MATRIZ PRODUCTO CORRESPONDIENTE, DETENIENDOSE.			$b_{i,j}$
6	SI SE DESEA COMPROBAR EL ÍNDICE, HACER		$x \rightleftarrows y$	i,j
7	REANUDAR CÁLCULO HACIENDO, PARA $i=1,\ldots,N$; $j=i,\ldots,N$ ($i \leq j$, POR SIMETRICA)		R/S	
8	VOLVER A 5.			
9	SI MOSTRADO EL ELEMENTO ÚLTIMO, b_{NN}, SE HACE		R/S	0,0000
10	PARA OTRA MULTIPLICACIÓN, IR A 3.			
11	PARA OTRO CASO, IR A 2.			

HP-34C PROGRAMA : MULTIPLICACION DE UNA MATRIZ NxN POR SU T.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL 1	6	g ISG	1	h RTN	6		1		6	
2	h SF 0	7	GTO 5	2	h LBL A	7		2		7	
3	GTO 0	8	RCL .8	3	GSB B	8		3		8	
4	h LBL 2	9	1	4	f R↑	9		4		9	
5	h CF 0	040	0	5	STO f I	110		5		180	
6	h LBL 0	1	÷	6	f R↑	1		6		1	
7	EEX	2	RCL .7	7	STO f (i)	2		7		2	
8	3	3	+	8		3		8		3	
9	÷	4	f FIX 1	9		4		9		4	
010	STO f I	5	h PSE	080		5		150		5	
1	g ISG	6	f FIX 9	1		6		1		6	
2	h LBL 3	7	RCL 0	2		7		2		7	
3	RCL f I	8	R/S	3		8		3		8	
4	STO .7	9	RCL .8	4		9		4		9	
5	h LBL 4	050	STO f I	5		120		5		190	
6	RCL .7	1	g ISG	6		1		6		1	
7	h FRAC	2	GTO 4	7		2		7		2	
8	f x⇄I	3	RCL .7	8		3		8		3	
9	g ISG	4	STO f I	9		4		9		4	
020	STO .8	5	g ISG	090		5		160		5	
1	CLX	6	GTO 3	1		6		1		6	
2	STO 0	7	CLX	2		7		2		7	
3	h LBL 5	8	f FIX 4	3		8		3		8	
4	RCL f I	9	h RTN	4		9		4		9	
5	RCL .7	060	h LBL B	5		130		5		200	
6	h F? 0	1	1	6		1		6		1	
7	x⇄y	2	−	7		2		7		2	
8	GSB B	3	4	8		3		8		3	
9	RCL f I	4	×	9		4		9		4	
030	RCL .8	5	+	100		5		170		5	
1	h F? 0	6	f x⇄I	1		6		1		6	
2	x⇄y	7	RCL f (i)	2		7		2		7	
3	GSB B	8	x⇄y	3		8		3		8	
4	×	9	f x⇄I	4		9		4		9	
5	STO + 0	070	g R↓	5		140		5		210	

Registros

0	b_{ij}	1	a_{11}	2	a_{12}	3	a_{13}	4	a_{14}	5	a_{21}	6	a_{22}
7	a_{23}	8	a_{24}	9	a_{31}	.0	a_{32}	.1	a_{33}	.2	a_{34}	.3	a_{41}
.4	a_{42}	.5	a_{43}	.6	a_{44}	.7	i	.8	j	.9		I	k, índice

Etiquetas

1: MULTIPLICACION DE A'A.

2: ID. AA' ; A: ALMACENAMIENTO DE a_{ij}.

B: DIRECCIONAMIENTO DE a_{ij}

0, 3, 4 Y 5.

Flags

0: SELECCIONA A'A Ó AA'.

Modo angular

Notacion

FIX 1 Y FIX 9

FACTORIZACION Y DETERMINANTE DE UNA MATRIZ CUADRADA

1.- Objeto.

Dada una matriz cuadrada $A=(a_{ij})$, de orden $N\leqslant 4$, tal que $a_{ii}\neq 0$ para cualquier i, el siguiente programa obtiene su factorización ortogonal y calcula su determinante.

De esta forma resulta $A = L\ U$, siendo:

$U = (u_{ij})$ la matriz reducida NxN triangular superior, con $u_{ii}\neq 1$ en general y $u_{ij}= 0$ para todo $i > j$.

$L = (l_{ij})$ la matriz de razones NxN triangular inferior, con $l_{ii}= 1$ para todo i, y con $l_{ij}= 0$ para todo $i < j$. El programa calculará los elementos $i > j$ y la matriz se deberá completar con 1 en la diagonal principal.

Por otra parte se simplifica notablemente el cálculo del determinante ya que se cumple: $DET = \prod_{i=1}^{N} u_{ii}$.

Si A es la matriz de un sistema, $A\ x = B$, y se extiende la reducción al miembro de la derecha de forma que $B = L\ C$, el — sistema equivalente $U\ x = C$ resulta de más facil solución al ser $u_{ij}= 0$ para $i > j$.

2.- Método.

Se emplea la reducción de Gauss, que puede especificarse en el algoritmo:

Hacer, para k = 1, 2, ..., N-1

- $m_{ik}= a_{ik}/a_{kk}$, i = k+1, ..., N
- Hacer, para i = k+1, ..., N
  - $a_{ij}= a_{ij} - m_{ik}a_{kj}$, j = 1, ..., N

Terminado el algoritmo hacer $u_{ij}= a_{ij}$ y $l_{ij}= m_{ij}$ quedando almacenados tales elementos en lugar de la matriz A.

Así, pues, los elementos empleados como pivotes son los de la diagonal principal. Al no emplearse ninguna estrategia de pivoteo, se hace necesaria la condición $a_{ii}\neq 0$ para evitar detenciones por operación ilegal. Si no se cumpliese, el usuario deberá alterar el orden de columnas y/o filas para lograrlo.

Tras la reducción, el determinante es facilmente calculable por el producto de los elementos de la diagonal principal.

3.- Observaciones.

Por el poco espacio de programa disponible no es posible almacenar los elementos a_{ij} automáticamente. La subrutina A direcciona tales elementos a partir de i, j, y mediante STO f (i) se almacenan. Tampoco se puede utilizar la capacidad del registro I por los pasos de programa necesarios para su preparación.

4.- Ejemplo.

Sea la matriz 4x4:

$$A = \begin{bmatrix} 5 & 0 & 2 & 3 \\ 2 & 4 & 1 & 1 \\ 1 & 3 & 10 & 2 \\ 0 & 7 & 5 & 6 \end{bmatrix}$$

Tras 4; STO 0 hacemos, siguiendo las instrucciones:

1; ENTER↑ ; A; 5; STO f (i)

2; ENTER↑ ; 1; A; 0; STO f (i)

3; ENTER↑ ; 1; A; 2; STO f (i)

Etc.

Arrancando el programa con B, muestra, tras 1 m 34 s de ejecución: DET = 1.056,0000. Extrayendo los elementos con A se obtiene:

$$U = \begin{bmatrix} 5,0000 & 0,0000 & 2,0000 & 3,0000 \\ & 4,0000 & 0,2000 & -0,2000 \\ & & 9,4500 & 1,5500 \\ & \bigcirc & & 5,5873 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & & \bigcirc & \\ 0,4000 & 1 & & \\ 0,2000 & 0,7500 & 1 & \\ 0,0000 & 1,7500 & 0,4921 & 1 \end{bmatrix}$$

Evidentemente los 1 de la diagonal principal no están almacenados ya que su lugar lo ocupan los elementos u_{ii} al estar ambas matrices superpuestas.

HP-34C

TITULO: FACTORIZACIÓN Y DETERMINANTE DE UNA MATRIZ CUADRADA.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA			
2	ALMACENAR ORDEN ($N \leq 4$)	N	STO 0	
3	ALMACENAR ELEMENTOS DE LA MATRIZ HACIEN-			
	DO, PARA $i = 1, \ldots, N$; $j = 1, \ldots, N$	j	ENTER ↑	
		i	A	CONT. ANT.
		a_{ij}	STO f(i)	
4	INICIAR CÁLCULO		B	DET
5	OBTENER ELEMENTOS DE LA MATRIZ REDUCIDA			
	HACIENDO, PARA $i = 1, \ldots, N$; $j = i, \ldots, N$	j	ENTER ↑	
	($i \leq j$)	i	A	u_{ij}
6	OBTENER ELEMENTOS DE LA MATRIZ TRIANGULAR			
	INFERIOR (MATRIZ DE RAZONES) HACIENDO, PARA			
	$i = 2, \ldots, N$; $j = 1, \ldots, i-1$ ($i > j$)	j	ENTER ↑	
		i	A	l_{ij}
7	OBTENER VALOR DEL DETERMINANTE		RCL 1	DET
8	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: FACTORIZACION Y DETERMINANTE DE MATR. NxN

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	RCL 1	1		6		1		6	
2	4	7	GSB A	2		7		2		7	
3	x	8	g R↓	3		8		3		8	
4	+	9	STO − (i)	4		9		4		9	
5	1	040	1	5		110		5		180	
6	−	1	STO − 2	6		1		6		1	
7	STO f I	2	RCL 3	7		2		7		2	
8	CLX	3	RCL 2	8		3		8		3	
9	RCL f(i)	4	f x>y	9		4		9		4	
010	h RTN	5	GTO 3	080		5		150		5	
1	h LBL B	6	1	1		6		1		6	
2	1	7	STO − 1	2		7		2		7	
3	STO 3	8	RCL 3	3		8		3		8	
4	h LBL 1	9	RCL 1	4		9		4		9	
5	RCL 0	050	f x>y	5		120		5		190	
6	STO 1	1	GTO 2	6		1		6		1	
7	h LBL 2	2	1	7		2		7		2	
8	RCL 3	3	STO + 3	8		3		8		3	
9	ENTER↑	4	RCL 0	9		4		9		4	
020	GSB A	5	RCL 3	090		5		160		5	
1	RCL 3	6	f x≠y	1		6		1		6	
2	RCL 1	7	GTO 1	2		7		2		7	
3	GSB A	8	1	3		8		3		8	
4	g R↓	9	STO 1	4		9		4		9	
5	STO ÷ (i)	060	h LBL 4	5		130		5		200	
6	RCL 0	1	RCL 3	6		1		6		1	
7	STO 2	2	ENTER↑	7		2		7		2	
8	h LBL 3	3	GSB A	8		3		8		3	
9	RCL 3	4	STO × 1	9		4		9		4	
030	GSB A	5	1	100		5		170		5	
1	RCL 3	6	STO − 3	1		6		1		6	
2	RCL 1	7	RCL 3	2		7		2		7	
3	GSB A	8	g x>0	3		8		3		8	
4	x	9	GTO 4	4		9		4		9	
5	RCL 2	070	RCL 1	5		140		5		210	

Registros

0 N	1 i, DET	2 j	3 k	4 a_{11}, u_{11}	5 a_{12}, u_{12}	6 a_{13}, u_{13}
7 a_{14}, u_{14}	8 a_{21}, l_{21}	9 a_{22}, u_{22}	.0 a_{23}, u_{23}	.1 a_{24}, u_{24}	.2 a_{31}, l_{31}	.3 a_{32}, l_{32}
.4 a_{33}, u_{33}	.5 a_{34}, u_{34}	.6 a_{41}, l_{41}	.7 a_{42}, l_{42}	.8 a_{43}, l_{43}	.9 a_{44}, u_{44}	I INDICE

Etiquetas

A: DIRECCIONAMIENTO DE ELEMENTOS.
B: CÁLCULO
1, 2, 3 y 4.

Flags

Modo angular

Notacion

FACTORIZACION Y DETERMINANTE DE UNA MATRIZ SIMETRICA

1.- Objeto.

Dada una matriz simétrica $A=(a_{ij})$, de orden $N \leqslant 5$, siendo evidentemente $a_{ij}=a_{ji}$ y tal que $a_{ii} \neq 0$ para cualquier i, este programa obtiene su factorización ortogonal y calcula su determinante.

De esta forma resulta A = L U, siendo:

$U = (u_{ij})$ la matriz reducida NxN triangular superior, con $u_{ii} \neq 1$ en general y $u_{ij} = 0$ para todo $i > j$. U es la matriz calculada por el programa.

$L = (l_{ij})$ la matriz de razones NxN triangular inferior, -- con $l_{ii} = 1$ para todo i, y con $l_{ij} = 0$ para todo $i < j$. Si bien el programa no calcula esta matriz, es facilmente obtenible ya que, por la simetría de A, se cumplirá $l_{ij} = u_{ji}/u_{jj}$ para todo $i > j$.

El cálculo del determinante se efectúa facilmente ya que $DET = \prod_{i=1}^{N} u_{ii}$.

También es extensible a este caso lo dicho en el programa anterior referente a la simplificación en la resolución de sistemas.

2.- Método.

Es idéntico al del programa anterior, pero aquí no se almacenan los elementos m_{ik}. La economía que supone la simetría de A permite aumentar en una unidad el orden máximo posible, llegando -- así a 5.

3.- Observaciones.

Es válido lo dicho en el anterior. La subrutina A se ha -- modificado adaptándola a los elementos a_{ij}, $i \leqslant j$. En evitación de - errores y para mayor comodidad, es indistinto el orden en que se ingresen i,j.

4.- Ejemplo.

Sea la matriz:

$$A = \begin{bmatrix} 8 & -2 & 3 & 0 & 4 \\ -2 & 10 & 5 & -4 & 2 \\ 3 & 5 & 6 & 0 & -1 \\ 0 & -4 & 0 & -7 & 1 \\ 4 & 2 & -1 & 1 & 5 \end{bmatrix}$$

Después de 5; STO 0, almacenamos los elementos $i < j$ haciendo:

1; ENTER↑ ; A; 8; STO f (i)
2; ENTER↑ ; 1; A; 2; CHS; STO f (i)
3; ENTER↑ ; 1; A; 3; STO f (i)
4; ENTER↑ ; 1; A; 0; STO f (i)
5; ENTER↑ ; 1; A; 4; STO f (i)
2; ENTER↑ ; A; 10; STO f (i)
3; ENTER↑ ; 2; A; 5; STO f (i)
Etc.

Iniciando con B, tras 3 m 20 s, muestra DET = 5.352,0000.
Recuperando los elementos reducidos, podemos formar la matriz:

$$U = \begin{bmatrix} 8,0000 & -2,0000 & 3,0000 & 0,0000 & 4,0000 \\ & 9,5000 & 5,7500 & -4,0000 & 3,0000 \\ & & 1,3947 & 2,4211 & -4,3158 \\ & 0 & & -12,8868 & 9,7547 \\ & & & & -3,9180 \end{bmatrix}$$

Para formar la matriz L haríamos:

1; ENTER↑ ; 2; A	-2,0000	
1; ENTER↑ ; A	8,0000	
/	-0,2500	(l_{21})
1; ENTER↑ ; 3; A	3,0000	
1; ENTER↑ ; A	8,0000	
/	0,3750	(l_{31})

Etc.

Obtenemos así:

$$L = \begin{bmatrix} 1 & & & 0 & \\ -0,2500 & 1 & & & \\ 0,3750 & 0,6053 & 1 & & \\ 0,0000 & -0,4211 & 1,7358 & 1 & \\ 0,5000 & 0,3158 & -3,0943 & -0,7570 & 1 \end{bmatrix}$$

HP-34C

TITULO: FACTORIZACION Y DETERMINANTE DE UNA MATRIZ SIMETRICA.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR ORDEN, $(N \leq 5)$	N	STO 0	
3	ALMACENAR ELEMENTOS DE LA MATRIZ HACIEN-			
	DO, PARA $i = 1, \ldots, N$; $j = i, \ldots, N$ $(i \leq j)$	j	ENTER ↑	
	(LOS ELEMENTOS $i > j$ NO SON NECESARIOS POR	i	A	CONT. ANT.
	SER SIMETRICA).	a_{ij}	STO $f(i)$	
4	INICIAR CÁLCULO		B	DET
5	OBTENER ELEMENTOS DE LA MATRIZ REDUCIDA HA-			
	CIENDO, PARA $i = 1, \ldots, N$; $j = i, \ldots, N$ $(i \leq j)$	j	ENTER ↑	
		i	A	u_{ij}
6	OBTENER VALOR DEL DETERMINANTE		RCL 1	DET
7	PARA UN NUEVO CASO, IR A 2.			
	NOTA:			
	SI FUESE NECESARIO CALCULAR LA MATRIZ DE			
	RAZONES, SE PODRÁ OBTENER HACIENDO, PARA			
	$i = 2, \ldots, N$; $j = 1, \ldots, i-1$ $(i > j)$	j	ENTER ↑	
		i	A	u_{ji}
		j	ENTER ↑	
			A	u_{jj}
			÷	l_{ij}

HP-34C PROGRAMA: FACTORIZ. Y DETERMINANTE DE UNA MATR. SIMETR.

N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION
001	h LBL A	6	×	1	RCL 3	6		1		6	
2	f x>y	7	RCL 3	2	g x>0	7		2		7	
3	x⇄y	8	ENTER↑	3	GTO 4	8		3		8	
4	1	9	GSB A	4	RCL 1	9		4		9	
5	−	040	÷	5	h RTN	110		5		180	
6	g	1	RCL 2	6		1		6		1	
7	x⇄y	2	RCL 1	7		2		7		2	
8	−	3	GSB A	8		3		8		3	
9	h LSTX	4	g R↓	9		4		9		4	
010	×	5	STO − (i)	080		5		150		5	
1	2	6	1	1		6		1		6	
2	÷	7	STO + 2	2		7		2		7	
3	+	8	RCL 0	3		8		3		8	
4	3	9	RCL 2	4		9		4		9	
5	+	050	f x≤y	5		120		5		190	
6	STO f I	1	GTO 3	6		1		6		1	
7	CLX	2	CLX	7		2		7		2	
8	RCL f(i)	3	RCL 1	8		3		8		3	
9	h RTN	4	f x≠y	9		4		9		4	
020	h LBL B	5	GTO 2	090		5		160		5	
1	1	6	1	1		6		1		6	
2	STO 3	7	STO + 3	2		7		2		7	
3	h LBL 1	8	RCL 0 .	3		8		3		8	
4	STO 1	9	RCL 3	4		9		4		9	
5	h LBL 2	060	f x≠y	5		130		5		200	
6	1	1	GTO 1	6		1		6		1	
7	STO + 1	2	1	7		2		7		2	
8	RCL 1	3	STO 1	8		3		8		3	
9	STO 2	4	h LBL 4	9		4		9		4	
030	h LBL 3	5	RCL 3	100		5		170		5	
1	RCL 3	6	ENTER↑	1		6		1		6	
2	GSB A	7	GSB A	2		7		2		7	
3	RCL 1	8	STO × 1	3		8		3		8	
4	RCL 3	9	1	4		9		4		9	
5	GSB 4	070	STO − 3	5		140		5		210	

Registros

0	N	1	i, DET	2	j	3	k	4	a_{11}, u_{11}	5	a_{12}, u_{12}	6	a_{13}, u_{13}
7	a_{14}, u_{14}	8	a_{15}, u_{15}	9	a_{22}, u_{22}	.0	a_{23}, u_{23}	.1	a_{24}, u_{24}	.2	a_{25}, u_{25}	.3	a_{33}, u_{33}
.4	a_{34}, u_{34}	.5	a_{35}, u_{35}	.6	a_{44}, u_{44}	.7	a_{45}, u_{45}	.8	a_{55}, u_{55}	.9		I	INDICE

Etiquetas

A : DIRECCIONAMIENTO DE a_{ij}.

B : CÁLCULO

1, 2, 3 Y 4.

Flags

Modo angular

Notacion

SOLUCION DE DOS ECUACIONES SIMULTANEAS CON COEFICIENTES COMPLEJOS

1.- Objeto.

Dado un sistema de dos ecuaciones simultáneas con coeficientes complejos, y expresado en la forma:

$$v_1 x_1 + v_2 x_2 = w_5$$
$$v_3 x_1 + v_4 x_2 = w_6$$

el siguiente programa calcula las raíces x_1 y x_2, que serán, en general, complejas.

2.- Método.

Para este propósito se emplea la regla de Cramer. Toda la aritmética ha de hacerse con complejos. Se prevé que los datos se den en coordenadas polares.

Así, pues, se calcula:

$$x_1 = \frac{\begin{vmatrix} w_5 & v_2 \\ w_6 & v_4 \end{vmatrix}}{\Delta} \qquad x_2 = \frac{\begin{vmatrix} v_1 & w_5 \\ v_3 & w_6 \end{vmatrix}}{\Delta} \qquad \Delta = \begin{vmatrix} v_1 & v_2 \\ v_3 & v_4 \end{vmatrix}$$

3.- Observaciones.

Ya que se ha de emplear reiteradas veces, la subrutina 2 efectúa la multiplicación de complejos en coordenadas polares.

El almacenamiento de los coeficientes se efectúa con la entrada 0, que termina calculando el determinante del sistema.

4.- Ejemplo.

Sea el sistema:

$$8\underline{|30}\ x_1 + 1\underline{|-15}\ x_2 = 10\underline{|60}$$
$$3\underline{|-30}\ x_1 + 12\underline{|125}\ x_2 = 2\underline{|-45}$$

Después de ajustar el modo angular (g DEG) iniciamos cargando los coeficientes:

GSB 0	0,0000
30; ENTER↑ ; 8; R/S..........	1,0000
-15; ENTER↑ ; 1; R/S..........	2,0000
-30; ENTER↑ ; 3; R/S..........	3,0000

125; ENTER ↑ ; 12; R/S............ 4,0000

60; ENTER ↑ ; 10; R/S............ 5,0000

-45; ENTER ↑ ; 2; R/S............ 0,0000

Para calcular las raíces, haremos:

A .... 1,2230 (mod. x_1)
x⇄y .... -330,2644 (arg. x_1)

B .... 0,2210 (mod. x_2)
x⇄y .... -273,2121 (arg. x_2)

Para conocer el determinante del sistema haremos:

RCL 0 .... 98,8244 (mod. Δ)
RCL 1 .... 154,4051 (arg. Δ)

HP-34C

TITULO: SOLUCION DE 2 ECUACIONES SIMULTANEAS CON COEFICIENTES COMPLEJOS.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	AJUSTAR MODO ANGULAR ADECUADO.			
3	INICIAR		GSB 0	0
4	INGRESAR COEFICIENTES HACIENDO	V_1 (arg)	ENTER↑	
		V_1 (mod)	R/S	1
	(EL SISTEMA SE CONSIDERA EXPRESADO:	V_2 (arg)	ENTER↑	
	$V_1 x_1 + V_2 x_2 = W_5$	V_2 (mod)	R/S	2
	$V_3 x_1 + V_4 x_2 = W_6$).	V_3 (arg)	ENTER↑	
		V_3 (mod)	R/S	3
		V_4 (arg)	ENTER↑	
		V_4 (mod)	R/S	4
		W_5 (arg)	ENTER↑	
		W_5 (mod)	R/S	5
		W_6 (arg)	ENTER↑	
		W_6 (mod)	R/S	0
5	CALCULAR LA 1ª RAIZ		A	x_1 (mod)
			$x \rightleftarrows y$	x_1 (arg)
6	CALCULAR LA 2ª RAIZ		B	x_2 (mod)
			$x \rightleftarrows y$	x_2 (arg)
7	PARA REPETIR NUEVAMENTE LOS CALCULOS, IR A			
	5 ó 6.			
8	PARA CONOCER EL DETERMINANTE DEL SISTEMA		RCL 0	Δ (mod)
			RCL 1	Δ (arg)
9	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: SOL. A 2 ECUAC. SIMULT. CON COEF. COMPLEJOS.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL 0	6	g R↓	1	RCL 9	6		1		6	
2	4	7	STO − 1	2	RCL 8	7		2		7	
3	.	8	RCL 1	3	h LBL 3	8		3		8	
4	0	9	RCL 0	4	GSB 2	9		4		9	
5	1	040	g →P	5	STO − 2	110		5		180	
6	5	1	STO 0	6	g R↓	1		6		1	
7	STO f I	2	g R↓	7	STO − 3	2		7		2	
8	h LBL 1	3	STO 1	8	RCL 3	3		8		3	
9	RCL f I	4	CLX	9	RCL 2	4		9		4	
010	h INT	5	h RTN	080	g →P	5		150		5	
1	4	6	h LBL A	1	RCL 0	6		1		6	
2	−	7	RCL .3	2	÷	7		2		7	
3	2	8	RCL .2	3	x⇄y	8		3		8	
4	÷	9	RCL .1	4	RCL 1	9		4		9	
5	R/S	050	RCL .0	5	−	120		5		190	
6	STO f (i)	1	GSB 2	6	x⇄y	1		6		1	
7	g ISG	2	STO 2	7	h RTN	2		7		2	
8	g R↓	3	g R↓	8	h LBL 2	3		8		3	
9	STO f (i)	4	STO 3	9	x⇄y	4		9		4	
020	g ISG	5	RCL 7	090	g R↓	5		160		5	
1	GTO 1	6	RCL 6	1	×	6		1		6	
2	RCL 5	7	RCL .5	2	g R↓	7		2		7	
3	RCL 4	8	RCL .4	3	+	8		3		8	
4	RCL .1	9	GTO 3	4	f R↑	9		4		9	
5	RCL .0	060	h LBL B	5	f →R	130		5		200	
6	GSB 2	1	RCL 5	6	h RTN	1		6		1	
7	STO 0	2	RCL 4	7		2		7		2	
8	g R↓	3	RCL .5	8		3		8		3	
9	STO 1	4	RCL .4	9		4		9		4	
030	RCL 7	5	GSB 2	100		5		170		5	
1	RCL 6	6	STO 2	1		6		1		6	
2	RCL 9	7	g R↓	2		7		2		7	
3	RCL 8	8	STO 3	3		8		3		8	
4	GSB 2	9	RCL .3	4		9		4		9	
5	STO − 0	070	RCL .2	5		140		5		210	

Registros

0	Δ (mod)	1	Δ (arg)	2	Utiliz.	3	Utiliz.	4	V_1 (mod)	5	V_1 (arg)	6	V_2 (mod)
7	V_2 (arg)	8	V_3 (mod)	9	V_3 (arg)	.0	V_4 (mod)	.1	V_4 (arg)	.2	W_5 (mod)	.3	W_5 (arg)
.4	W_6 (mod)	.5	W_6 (arg)	.6		.7		.8		.9		I	i

Etiquetas

A : CALCULO 1ª RAIZ ; B : CALCULO 2ª RAIZ.

0 : ALMACENAMIENTO DE COEF. Y CALCULO DEL DETERMINANTE.

1, 2 y 3.

Flags

Modo angular

Notacion

SUMA DE PRODUCTOS DE ORDEN N DE M NUMEROS REALES

1.- Objeto.

Dada la serie de números reales x_1, x_2, ...x_M, el siguiente programa obtiene la suma de productos de orden N, es decir, la suma de los posibles productos que se pueden formar tomando N factores de entre los M números.

Por necesidades de almacenamiento, se ha de cumplir que -- $N + M \leq 16$, y por la filosofía del programa, $N \leq M$.

2.- Método.

Los posibles productos de N factores tomados de la serie de M números se forman mediante las combinaciones, sin repetición, (de ahí que $N \leq M$), de tales números de orden N. Es decir:

$$S_N = \sum_{k=1}^{l} p_k, \text{ con } l = \binom{M}{N} \text{ y } p_k = x_{i_1} x_{i_2} x_{i_3} \dots x_{i_N}$$

con $i_1 = 1,\dots,M$; $i_2 = 1,\dots,M$; ... ; $i_N = 1,\dots,M$;
e $i_1 \neq i_2 \neq \dots \neq i_N$.

3.- Observaciones.

El programa sigue el método expuesto, por lo que los índices i_k se generan en el proceso de cálculo. Ya que se requiere su -- almacenamiento, y teniendo disponibles sólo 16 registros, se establece la condición $N + M \leq 16$.

En el caso frecuente en que N = M, la condición se traduce en $M \leq 8$. Este es el caso en que se desea formar un polinomio conocidas sus raíces.

4.- Ejemplo.

1) Sean $x_1 = x_2 = x_3 = x_4 = 1$, es decir, M = 4. Las acumulaciones de productos de orden 1, 2, 3 y 4 nos darían los números combinatorios $\binom{4}{1}$, $\binom{4}{2}$, $\binom{4}{3}$, y $\binom{4}{4}$.

Efectivamente, tras 4; STO.9 y 1;STO 1; STO 2; STO 3; STO 4, obtenemos:

1; A	4,0000	(18 s)
2; A	6,0000	(44 s)

3; A .... 4,0000 (47 s)

4; A .... 1,0000 (22 s)

2) Queremos formar ahora un polinomio cuyas raíces sean $x_1 = 2$, $x_2 = -3$, $x_3 = 1$, $x_4 = 5$ y $x_5 = -4$, siendo por ello de 5º grado.

Se deberá cumplir $p(x) = (x-x_1)(x-x_2)\ldots(x-x_5)$ y de este producto se deduce que los coeficientes del polinomio son precisamente las acumulaciones S de los productos de los valores $-x_i$. Se puede establecer, en general, que $a_i = S_{N-i}$ siendo N el orden del polinomio.

Así, pues, tras 5; STO.9 y -2; STO 1; 3; STO 2; -1; STO 3; -5; STO 4; y 4; STO 5 obtenemos los coeficientes:

$a_0 = S_5$, luego 5; A .... -120,0000 (27 s)
$a_1 = S_4$, luego 4; A .... 134,0000 (1 m 15 s)
$a_2 = S_3$, luego 3; A .... 13,0000 (1 m 40 s)
$a_3 = S_2$, luego 2; A .... -27,0000 (1 m 9 s)
$a_4 = S_1$, luego 1; A .... -1,0000 (21 s)

El último coeficiente ha de ser $a_5 = 1$, según se desprende de la forma multiplicativa. Queda así:

$$p(x) = -120 + 134\ x + 13\ x^2 - 27\ x^3 - x^4 + x^5.$$

HP-34C

TITULO: SUMA DE PRODUCTOS DE ORDEN N, DE M NUMEROS REALES.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR NUMERO DE ELEMENTOS	M	STO . 9	
3	ALMACENAR ELEMENTOS HACIENDO, PARA			
	$K = 1, \ldots, M$	X_K	STO [K]	
4	INGRESAR ORDEN E INICIAR ($N \leq M$; $N+M \leq 16$)	N	A	SUMA
5	PARA OTRO ORDEN, IR A 4.			
6	PARA OTRO CASO, IR A 2.			

HP-34C PROGRAMA : SUMA DE PRODUCTOS DE ORDEN N.

N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION
001	h LBL A	6	RCL f(i)	1		6		1		6	
2	STO .8	7	f x⇄I	2		7		2		7	
3	1	8	RCL f(i)	3		8		3		8	
4	STO .7	9	x⇄y	4		9		4		9	
5	GSB 0	040	f x⇄I	5		110		5		180	
6	CLX	1	g R↓	6		1		6		1	
7	STO f(i)	2	×	7		2		7		2	
8	STO 0	3	g DSE	8		3		8		3	
9	h LBL 1	4	GTO 3	9		4		9		4	
010	1	5	STO + 0	080		5		150		5	
1	STO + (i)	6	h LBL 4	1		6		1		6	
2	RCL f(i)	7	RCL .9	2		7		2		7	
3	RCL .7	8	RCL .7	3		8		3		8	
4	RCL .8	9	+	4		9		4		9	
5	f x=y	050	RCL .8	5		120		5		190	
6	GTO 2	1	−	6		1		6		1	
7	CLX	2	RCL .7	7		2		7		2	
8	1	3	GSB 0	8		3		8		3	
9	+	4	f x≠y	9		4		9		4	
020	STO .7	5	GTO 1	090		5		160		5	
1	GSB 0	6	RCL .7	1		6		1		6	
2	g R↓	7	1	2		7		2		7	
3	STO f(i)	8	−	3		8		3		8	
4	GTO 1	9	STO .7	4		9		4		9	
5	h LBL 2	060	g x>0	5		130		5		200	
6	RCL .9	1	GTO 4	6		1		6		1	
7	+	2	RCL 0	7		2		7		2	
8	RCL .9	3	h RTN	8		3		8		3	
9	EEX	4	h LBL 0	9		4		9		4	
030	3	5	RCL .9	100		5		170		5	
1	÷	6	+	1		6		1		6	
2	+	7	STO f I	2		7		2		7	
3	STO f I	8	CLX	3		8		3		8	
4	1	9	RCL f(i)	4		9		4		9	
5	h LBL 3	070	h RTN	5		140		5		210	

Registros

0	SUMA	1	x_1	2	x_2	3	x_3	4	x_4	5	x_5	6	x_6
7	x_7	8	x_8	9	I_1	.0	I_2	.1	I_3	.2	I_4	.3	I_5
.4	I_6	.5	I_7	.6	I_8	.7	K	.8	N	.9	M	I	i

Etiquetas

A : CALCULA LA SUMA.

0 : DIRECCIONAMIENTO DE I_k

1, 2, 3 y 4.

Flags

Modo angular

Notacion

ALGORITMO DE HORNER

1.- Objeto.

Dado el polinomio $y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$, $n \leqslant 16$, -- el siguiente programa proporciona:

1) División del polinomio entre el binomio $x - z$ (Sb. A).

2) Obtención de $y = f(z)$ (Sb. A).

3) Cambio de ejes $x = x' + z$, resultando un polinomio en x' de grado n (Sb. B).

4) Obtención de los coeficientes de Taylor para $x = z$ --- (Sb. B).

5) Eliminación del término x^{n-1} mediante la traslación -- $x' = x + a_{n-1}/n$, cuando $a_n = 1$ (Sb. B).

2.- Método.

Se emplea el conocido algoritmo de Horner, que en su forma general se puede especificar:

Tomar $j = 0, 1, \ldots, n$.
Hacer, para $i = (n-1), (n-2), \ldots, j$
$a_i = a_i + z\, a_{i+1}$

Tal como está especificado, los coeficientes modificados quedan en el mismo lugar que los iniciales. En realidad, tras la -- primera pasada, correspondiente a $j = 0$, el grado del polinomio queda reducido en una unidad, por lo que los coeficientes deberían --- trasladarse un registro. En la versión utilizada para nuestro propósito se hace así. En realidad se empléa $i = j, j-1, j-2, \ldots, 0$, y $j = n-1, n-2, \ldots, 0$. Después de cada j se efectúa una rotación de los coeficientes de forma que $a_{n-1} = a_n$, $a_{n-2} = a_{n-1}$, etc., y el resto, correspondiente al nuevo valor de a_0, se almacena en a_n.

Para los objetivos 1) y 2) sólo es necesaria una primera - pasada del algoritmo, correspondiente a $j = n$ en nuestra versión.

3.- Observaciones.

En el caso de un sólo ciclo se emplea la entrada A. Para el resto de utilizaciones, en que se requiere el algoritmo completo, se emplea la entrada B. La subrutina LBL 4 es la que efectúa la multiplicación encajada de los coeficientes subindicados desde el -- valor actual de j hasta 0. La subrutina termina (LBL 5) con la rotación de coeficientes.

4.- <u>Ejemplos</u>.

Sea el polinomio

$$p(x) = 60 - 32\,x - 71\,x^2 + 40\,x^3 + 10\,x^4 - 8\,x^5 + x^6.$$

1) Se pretende dividirlo por el binomio x - 4; es decir, z = 4. Después de 6; STO.9 y de almacenar los coeficientes con --- 60; STO 0; 32; CHS; STO 1; 71; CHS; STO 2; 40; STO 3; Etc. haremos:

4; A .... -180 (12 s)

valor correspondiente al resto de la división. Recuperando los nuevos coeficientes tendremos:

RCL 0 .... -60,0000
RCL 1 .... -7,0000
... ...
RCL 6 .... -180,0000

pudiendose formar:

$$p(x)/(x-4) = -60 - 7\,x + 16\,x^2 - 6\,x^3 - 4\,x^4 + x^5 - 180/(x-4).$$

2) La sustitución x = 4 la hubiésemos resuelto de la misma forma. Cargado nuevamente, 4; A, nos daría -180, resto que equivale a p(4).

3) Queremos hacer ahora la traslación de ejes $x = x' + 3$, es decir, z = 3. Almacenamos nuevamente los coeficientes y hacemos ---- 3; B. Tras 1 m 1 s nos muestra el orden 6. Recuperados los nuevos -- coeficientes obtenemos:

$$p(x') = -\,80\,x - 116\,x^2 - 20\,x^3 + 25\,x^4 + 10\,x^5 + x^6.$$

4) Si quisiéramos conocer los coeficientes de Taylor para - x = 3, haríamos igual que antes, es decir 3; B, y tales coeficientes serían los obtenidos antes. Luego si

$$p(x+z) = p(z) + \frac{p'(z)}{\lfloor 1}x + \frac{p''(z)}{\lfloor 2}\,x^2 + \;\ldots =$$

$$= b_0 + b_1\,x + b_2\,x^2 + \;\ldots$$

el desarrollo de p(x + 3) es el mismo que el obtenido antes para $p(x')$.

5) Pretendemos ahora eliminar el término en x^5. Puesto que $a_6 = 1$, hemos de hacer la traslación:

$x' = x + a_{n-1}/n$; $x = x' - a_{n-1}/n$; luego $z = -a_{n-1}/n$.

En nuestro caso sería $z = 8/6 = 1,333333333$.

Almacenados los coeficientes del polinomio haríamos:

f FIX 9; 8; ENTER ; 6; /; B 6,000000000

Recuperando los nuevos coeficientes obtenemos:

RCL 0	-10,56241426	(b_0)
RCL 1	-14,32098769	(b_1)
RCL 2	53,44444445	(b_2)
RCL 3	-1,481481460	(b_3)
RCL 4	-16,66666667	(b_4)
RCL 5	-0,000000002	(b_5)
RCL 6	1,000000000	(b_6)

Los errores de redondeo, al no ser z una fracción exacta, ha producido b_5 algo menor que 0.

HP-34C

TITULO: ALGORITMO DE HORNER

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR GRADO DEL POLINOMIO ($N \leq 16$)	N	STO . 9	
3	ALMACENAR COEFICIENTES HACIENDO, PARA $k=$			
	$0, \ldots, N$	a_k	STO [k]	
4	INGRESAR TERMINO DE SUSTITUCION	z		
5	PARA UN SOLO CICLO, HACER		A	RESTO
6	PARA ALGORITMO COMPLETO, HACER		B	N
7	RECUPERAR NUEVOS COEFICIENTES HACIENDO,			
	PARA $k = 0, 1, \ldots, N$		RCL [k]	b_k
8	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA : ALGORITMO DE HORNER

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	g DSE	1		6		1		6	
2	STO .7	7	GTO 2	2		7		2		7	
3	RCL .9	8	h x⇄(i)	3		8		3		8	
4	STO .8	9	RCL .9	4		9		4		9	
5	GTO 4	040	f x⇄I	5		110		5		180	
6	h LBL B	1	g R↓	6		1		6		1	
7	STO .7	2	h x⇄(i)	7		2		7		2	
8	RCL .9	3	RCL f(i)	8		3		8		3	
9	STO .8	4	h RTN	9		4		9		4	
010	h LBL 3	5	h LBL 1	080		5		150		5	
1	GSB 4	6	STO + f(i)	1		6		1		6	
2	RCL .8	7	GTO 1	2		7		2		7	
3	1	8	h LBL 2	3		8		3		8	
4	−	9	h x⇄(i)	4		9		4		9	
5	STO .8	050	GTO 2	5		120		5		190	
6	g x≠0	1		6		1		6		1	
7	GTO 3	2		7		2		7		2	
8	GSB 5	3		8		3		8		3	
9	RCL .9	4		9		4		9		4	
020	h RTN	5		090		5		160		5	
1	h LBL 4	6		1		6		1		6	
2	RCL .8	7		2		7		2		7	
3	STO f I	8		3		8		3		8	
4	h LBL 1	9		4		9		4		9	
5	RCL f(i)	060		5		130		5		200	
6	RCL .7	1		6		1		6		1	
7	×	2		7		2		7		2	
8	g DSE	3		8		3		8		3	
9	GTO 1	4		9		4		9		4	
030	STO + 0	5		100		5		170		5	
1	h LBL 5	6		1		6		1		6	
2	RCL .9	7		2		7		2		7	
3	STO f I	8		3		8		3		8	
4	RCL f(i)	9		4		9		4		9	
5	h LBL 2	070		5		140		5		210	

Registros

0	a_0	1	a_1	2	a_2	3	a_3	4	a_4	5	a_5	6	a_6
7	a_7	8	a_8	9	a_9	.0	a_{10}	.1	Etc.	.2		.3	
.4		.5		.6		.7	z	.8	J	.9	N	I	I

Etiquetas

A : EFECTUA UN SOLO CICLO DEL ALGORITMO.

B : EJECUCION COMPLETA.

1, 2, 3, 4 y 5.

Flags

Modo angular

Notacion

ALGORITMO DE HORNER DE DOBLE FILA

1.- Objeto.

Dado el polinomio $p(x) = a_o + a_1 x + a_2 x^2 + \dots + a_n x^n$, con $n \leqslant 15$, y el trinomio $Q = x^2 - s x - p$, el siguiente programa -- calcula:

1) La división $p(x)/Q$, obteniendo el polinomio cociente entero $g(x)$, de grado $n - 2$, y el binomio resto $A_o + B_o x$, tal que $p(x) = Q\ g(x) + A_o + B_o x$. (Sb. A).

2) Obtiene el valor $p(x_o)$ para x_o complejo. Siendo ------- $x_o = u \pm i v$, y obteniendo $s = 2u$; $p = -(u^2 + v^2)$, el valor buscado será $p(x_o) = A_o + B_o x_o$. (Sb. B).

3) Para x_o real, bastará encontrar s y p que verifiquen -- $x_o^2 = s x_o + p$. Aplicando el algoritmo se tendrá:

$$p(x_o) = A_o + B_o x_o \qquad \text{(Sb. B)}$$

4) Dividiendo sucesivamente $p(x)$ entre Q se obtendrá la -- descomposición

$$p(x) = (A_o + B_o x) + (A_1 + B_1 x)Q + (A_2 + B_2 x)Q^2 + \dots$$

en la que los binomios $A_i + B_i x$ son los sucesivos restos. (Sb. B).

5) Considerando $A_o = A(s,p)$; $B_o = B(s,p)$; es decir, como funciones de s, p, se pueden deducir las distintas derivadas de A y B como sigue:

Denominando $A_p = \dfrac{\partial A}{\partial p}$; $A_s = \dfrac{\partial A}{\partial s}$; $B_p = \dfrac{\partial B}{\partial p}$; $B_s = \dfrac{\partial B}{\partial s}$

$$A_{pp} = \frac{\partial^2 A}{\partial p^2} \ ; \ A_{ps} = \frac{\partial^2 A}{\partial p \partial s}; \quad A_{ss} = \frac{\partial^2 A}{\partial s^2} \ ; \quad \text{Etc.}$$

resulta: $A_p = A_1$; $B_p = B_1$; $A_s = pB_1$; $B_s = sB_p + A_p$

$A_{pp}/2 = A_2$ $\qquad$ $B_{pp}/2 = B_2$

$A_{ps}/2 = pB_2 + B_1/2$ $\qquad$ $B_{ps}/2 = sB_2 + A_2 = C_2$

$A_{ss}/2 = pC_2$ $\qquad$ $B_{ss}/2 = sC_2 + pB_2 + B_1$

Etc. (Sb. B).

2.- Método.

Disponiendo la división como un esquema de Horner de doble fila, se deduce facilmente:

	a_n	a_{n-1}	a_{n-2}	a_{n-3}	.....	a_3	a_2	a_1	a_o
p	-	-	pa_n	pa'_{n-1}	.....	pa'_5	pa'_4	pa'_3	pa'_2
s	-	sa_n	sa'_{n-1}	sa'_{n-2}	.....	sa'_4	sa'_3	sa'_2	-
	a_n	a'_{n-1}	a'_{n-2}	a'_{n-3}	.....	a'_3	a'_2	a'_1	a'_o

Desde a'_2 a a_n serán los coeficientes del polinomio cociente de grado n-2. Los términos de cierre $a'_1 = B_o$ y $a'_o = A_o$ son los del binomio de resto.

Se puede establecer, pues, el siguiente algoritmo:

Tomar $a_n = a_n$; $a_{n+1} = a_{n+2} = \dots = 0$

Para i = (n-1), (n-2), ..., 1, hacer

$$a_i = a_i + a_{i+1}\, s + a_{i+2}\, p$$

Tomar $a_o = a_o + p\, a_2$; $A_o = a_o$; $B_o = a_1$

Aplicado reiteradas veces, disminuyendo cada vez el grado en dos unidades y rotando los coeficientes, obtendremos los restos A_o, B_o; A_1, B_1; etc. hasta obtener el último cociente de grado < 2. Los valores A_i, B_i quederán almacenados en lugar de los a_i y situados correctamente por la traslación a la que se les somete.

3.- Observaciones.

La rotación de coeficientes ha de hacerse en dos órdenes, por disminuir el grado en dos unidades. Esto lo realiza LBL 7, que emplea por dos veces la subrutina 5. Cuando el grado es par, el último resto carece del término en x, y sólo es necesario rotar una -- vez; de aquí las dos pruebas en los pasos 019 y 021.

4.- Ejemplos.

1) Sea el polinomio $p(x) = 28 + 5x + 3x^2 + x^3 + x^4$. Deseamos dividirlo entre $Q = 4 - 2x + x^2$.

Hacemos: 4; STO.9; 28; STO 0; 5; STO 1; 3; STO 2; 1; STO 3; STO 4.

2; STO.7; 4; CHS; STO.8

A 4,0000

Recuperando los coeficientes, obtenemos:

RCL 0	5,0000	(b_o)
RCL 1	3,0000	(b_1)
RCL 2	1,0000	(b_2)

RCL 3 .... 8,0000 (A_0)
RCL 4 .... 3,0000 (B_0)

Por lo que podremos formar:

$$p(x)/Q = 5 + 3x + x^2 + (8 + 3x)/Q.$$

2) Dado el mismo polinomio anterior, queremos conocer su valor para $x_0 = 1 + i\sqrt{3}$.

Formamos $s = 2u = 2$; $p = -(u^2 + v^2) = 4$. Son, pues, los mismos valores que en 1), por lo que repetiríamos los mismos pasos anteriores para obtener $A_0 = 8$ y $B_0 = 3$. Así, sería:

$$p(x_0) = A_0 + B_0 x_0 = 8 + 3(1 + i\sqrt{3}) = 11 + i3\sqrt{3}\ .$$

3) Sea ahora el polinomio

$$p(x) = 28 + 21x - 13x^2 + 13x^3 - 3x^4 + x^5$$

y deseamos descomponerlo en función de $Q = 4 - 2x + x^2$ mediante sucesivas divisiones. Haremos:

5; STO.9.

28; STO 0; 21; STO 1; 13; CHS; STO 2; 13; STO 3;
3; CHS; STO 4; 1; STO 5; 2; STO.7; 4; CHS; STO.8

B .... 5,0000 (N) (En 43 s)

RCL 0 .... 8,0000 (A_0)
RCL 1 .... 3,0000 (B_0)
RCL 2 .... 1,0000 (A_1)
RCL 3 .... 5,0000 (B_1)
RCL 4 .... 1,0000 (A_2)
RCL 5 .... 1,0000 (B_2)

Resultando, así:

$$p(x) = 8 + 3x + Q(1 + 5x) + Q^2(1 + x).$$

HP-34C

TITULO: ALGORITMO DE HORNER DE DOBLE FILA.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR GRADO DEL POLINOMIO ($N \leq 15$)	N	STO . 9	
3	ALMACENAR COEFICIENTES HACIENDO, PARA $k = 0, \ldots, N$	a_k	STO [k]	
4	ALMACENAR COEFICIENTES DEL FACTOR CUADRÁTICO	S	STO . 7	
		P	STO . 8	
5	PARA UN SOLO CICLO HACER		A	N
6	PARA ALGORITMO COMPLETO, HACER		B	N
7	RECUPERAR NUEVOS COEFICIENTES HACIENDO:			
7.1	SI SE EFECTUÓ UN SOLO CICLO (#A: UNA DIVISIÓN), HACER, PARA $k = 0, \ldots, N-2$		RCL [k]	b_k
	OBTENER EL BINOMIO RESTO HACIENDO		RCL [N-1]	A_0
			RCL [N]	B_0
7.2	SI SE EFECTUÓ EL ALGORITMO COMPLETO, OBTENER LOS RESTOS SUCESIVOS HACIENDO, PARA $k = 0, 2, 4, \ldots, N$		RCL [k]	$A_{k/2}$
	(SI N ES PAR SE CARECERÁ DEL ÚLTIMO $B_{k/2}$, O LO QUE ES IGUAL, SERÁ NULO)		RCL [k+1]	$B_{k/2}$
8	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: ALGORITMO DE HORNER DE DOBLE FILA.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	g R↓	1		6		1		6	
2	RCL .9	7	+	2		7		2		7	
3	STO .6	8	STO + (i)	3		8		3		8	
4	h LBL 4	9	f R↑	4		9		4		9	
5	STO f I	040	RCL f(i)	5		110		5		180	
6	RCL f(i)	1	GTO 2	6		1		6		1	
7	g DSE	2	h LBL 1	7		2		7		2	
8	GTO 1	3	RCL .7	8		3		8		3	
9	h RTN	4	x⇄y	9		4		9		4	
010	h LBL B	5	×	080		5		150		5	
1	RCL .9	6	STO + (i)	1		6		1		6	
2	STO .6	7	h LSTx	2		7		2		7	
3	h LBL 3	8	RCL f(i)	3		8		3		8	
4	GSB 4	9	h LBL 2	4		9		4		9	
5	RCL .6	050	g DSE	5		120		5		190	
6	2	1	GTO 2	6		1		6		1	
7	−	2	CLx	7		2		7		2	
8	STO .6	3	RCL .8	8		3		8		3	
9	g x>0	4	×	9		4		9		4	
020	GTO 3	5	STO + 0	090		5		160		5	
1	g x=0	6	h LBL 7	1		6		1		6	
2	GTO 5	7	GSB 5	2		7		2		7	
3	GTO 7	8	h LBL 5	3		8		3		8	
4	h LBL 6	9	RCL .9	4		9		4		9	
5	h x⇄(i)	060	STO f I	5		130		5		200	
6	GTO 6	1	RCL f(i)	6		1		6		1	
7	h LBL 2	2	h LBL 6	7		2		7		2	
8	x⇄y	3	g DSE	8		3		8		3	
9	RCL .8	4	GTO 6	9		4		9		4	
030	×	5	h x⇄(i)	100		5		170		5	
1	x⇄y	6	RCL .9	1		6		1		6	
2	RCL .7	7	f x⇄I	2		7		2		7	
3	x⇄y	8	x⇄y	3		8		3		8	
4	×	9	h x⇄(i)	4		9		4		9	
5	h LSTx	070	x⇄y	5		140		5		210	

Registros

0	a_0 , A_0	1	a_1 , B_0	2	a_2 , A_1	3	a_3 , B_1	4	a_4 , A_2	5	a_5 . B_2	6	a_6 . A_3
7	a_7 , B_3	8	ETC.	9		.0		.1		.2		.3	
.4		.5		.6	J	.7	S	.8	P	.9	N	I	I

Etiquetas

A : EJECUTA UN SOLO CICLO.

B : EJECUTA ALGORITMO COMPLETO

1, 2, 3, 4, 5, 6 Y 7.

Flags

Modo angular

Notacion

RECIPROCO DE UN DESARROLLO POLINOMICO

1.- Objeto.

Dado un desarrollo polinómico de la forma

$$P = 1 + a_1 x + a_2 x^2 + \ldots$$

el siguiente programa obtiene los coeficientes, hasta los 19 primeros, del desarrollo recíproco del anterior:

$$R = 1 + b_1 x + b_2 x^2 \quad \ldots$$

de forma que $P \cdot R = 1$, es decir, $R = P^{-1}$.

2.- Método.

Por definición se deberá cumplir $P\,R = 1$ y, por consiguiente

$$(1 + a_1 x + a_2 x^2 + \ldots)(1 + b_1 x + b_2 x^2 + \ldots) = 1$$

lo que supone que, para $n = 1, 2, \ldots$

$$b_n + a_1 b_{n-1} + a_2 b_{n-2} + \ldots + a_n = 0$$

obteniéndose la fórmula recurrente

$$b_n = - a_1 b_{n-1} - a_2 b_{n-2} - \ldots - a_n b_o \;, \text{ con } b_o = 1 \text{ y } n = 1, 2, \ldots$$

La expresión anterior puede sintetizarse en la siguiente:

$$b_i = - a_i - \sum_{j=1}^{i-1} a_j b_{i-j} \;, \qquad i = 1, 2, \ldots, n$$

siendo n aquí el número de coeficientes b_i deseados.

3.- Observaciones.

Hay disponibles 19 registros para almacenamiento de a_i y b_i. Cuando a_i es función de i, como ocurre en muchos casos prácticos, y es posible, pues, programar $a_i = f(i)$, los 19 registros quedan disponibles para b_i por lo que $N \leqslant 19$, siendo N el número de —coeficientes b_i deseados.

Cuando a_i deben ser explicitados y, por consiguiente, almacenados, los 19 disponibles han de ser repartidos entre a_i y b_i. Así, deberá cumplirse $N_a + N \leqslant 19$, siendo N_a el grado de P y N, como antes.

El programa presentado considera esta última circunstancia y por ello la subrutina A extrae de la memoria los valores a_i, debiéndose ingresar N_a como constante en el paso 049 (y 050 si tuviese dos dígitos).

En otro caso, cuando $a_i = f(i)$, en lugar de lo enlistado -- desde 049 al final, se programará esta función, siempre bajo LBL A.

4.- Ejemplos.

1) Se desea calcular los 8 primeros términos de la expansión:

$$-\frac{x}{\log (1-x)} = 1 + b_1x + b_2x^2 + \dots$$

La serie buscada es la recíproca de

$$P = -\frac{\log (1-x)}{x} = 1 + x/2 + x^2/3 + \dots + a_ix^i + \dots$$

Los coeficientes a_i vienen dados, pues, por

$$a_i = 1/(i+1) \ , \quad i = 1, 2, \dots$$

No usaremos la subrutina A enlistada, sino esta otra:

```
048   LBL A
049     1
050     +
051   h 1/x
052   h RTN
```

Bastará ahora hacer:

8; B	8,0000	(1 m 38 s)
f FIX 9		
RCL 1	-0,500000000	(b_1)
RCL 2	-0,083333333	(b_2)
RCL 3	-0,041666667	(b_3)
RCL 4	-0,026388889	(b_4)
RCL 5	-0,018750000	(b_5)
RCL 6	-0,014269180	(b_6)
RCL 7	-0,011367394	(b_7)
RCL 8	-0,009356537	(b_8)

2) Deseamos ahora obtener el desarrollo del recíproco de

$$P = 1 + x + 3x^2 - 5x^3$$

Hemos de emplear aquí la subrutina A presentada, ingresando 3 en el paso 049, por ser el grado de P. Para calcular los 8 primeros términos, haremos: 5; CHS; STO.9; 3; STO.8; 1; STO.7

8; B	8,0000	(2 m 6 s)
RCL 1	-1,0000	(b_1)
RCL 2	-2,0000	(b_2)
RCL 3	10,0000	(b_3)
RCL 4	-9,0000	(b_4)
RCL 5	-31,0000	(b_5)
RCL 6	108,0000	(b_6)
RCL 7	-60,0000	(b_7)
RCL 8	-419,0000	(b_8)

Nota: La base teórica del método y el primer ejemplo están tomados de "Computational Analysis with the HP-25 Pocket Calculator".- P.

HP-34C

TITULO: RECIPROCO DE UN DESARROLLO POLINOMICO.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA HASTA EL PASO 048.			
2	SI SE HA DE EXPLICITAR a_k, CARGAR LOS PA-			
	SOS RESTANTES, DESDE 049 AL FINAL, INGRE-			
	SANDO POR N_a EL ORDEN DEL POLINOMIO.			
2 bis	SI a_k ES IMPLICITO Y CALCULABLE EN FUNCION			
	DE k, NO UTILIZAR DESDE 049 EN ADELAN-			
	TE, PROGRAMANDO EN SU LUGAR LA OBTENCIÓN			
	DE a_k SUPONIENDO A k EN EL REGISTRO X			
	Y DEVOLVIENDO a_k TAMBIEN EN X. IR A 4.			
3	EN EL CASO DE a_k EXPLICITOS, ALMACENAR ES-			
	TOS HACIENDO	a_{N_a}	STO . 9	
		a_{N_a-1}	STO . 8	
		a_{N_a-2}	STO . 7	
		Etc.		
4.	INGRESAR NUMERO DE TERMINOS DEL RECI-			
	PROCO DESEADOS ($N+N_a \leq 19$) E INICIAR	N	B	N
5	RECUPERAR TERMINOS b_k HACIENDO, PARA			
	$k = 1, \ldots, N$		RCL k	b_k
6	PARA OTRO CASO, IR A 2.			

HP-34C PROGRAMA : RECIPROCO DE UN DESARROLLO POLINOMICO.

N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION
001	h LBL B	6	RCL f I	1		6		1		6	
2	EEX	7	1	2		7		2		7	
3	3	8	−	3		8		3		8	
4	÷	9	h INT	4		9		4		9	
5	1	040	h RTN	5		110		5		180	
6	+	1	h LBL 3	6		1		6		1	
7	STO f I	2	f x⇄I	7		2		7		2	
8	h LBL 1	3	RCL f (i)	8		3		8		3	
9	RCL f I	4	x⇄y	9		4		9		4	
010	h INT	5	f x⇄I	080		5		150		5	
1	GSB A	6	g R↓	1		6		1		6	
2	CHS	7	h RTN	2		7		2		7	
3	1	8	h LBL A	3		8		3		8	
4	STO 0	9	[Na]	4		9		4		9	
5	h LBL 2	050	x⇄y	5		120		5		190	
6	g R↓	1	f x>y	6		1		6		1	
7	RCL 0	2	GTO 4	7		2		7		2	
8	GSB A	3	−	8		3		8		3	
9	RCL 0	4	1	9		4		9		4	
020	RCL f I	5	9	090		5		160		5	
1	h INT	6	−	1		6		1		6	
2	f x=y	7	GTO 3	2		7		2		7	
3	GTO 1	8	h LBL 4	3		8		3		8	
4	−	9	×	4		9		4		9	
5	GSB 3	060	CLX	5		130		5		200	
6	×	1	h RTN	6		1		6		1	
7	−	2		7		2		7		2	
8	1	3		8		3		8		3	
9	STO + 0	4		9		4		9		4	
030	GTO 2	5		100		5		170		5	
1	h LBL 1	6		1		6		1		6	
2	f R↑	7		2		7		2		7	
3	STO f (i)	8		3		8		3		8	
4	g ISG	9		4		9		4		9	
5	GTO 1	070		5		140		5		210	

Registros

0	J	1	b_1	2	b_2	3	b_3	4	Etc.	5		6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6	Etc.	.7	a_{n-2}	.8	a_{n-1}	.9	a_n	I	I

Etiquetas

A : CALCULO O DIRECCIONAMIENTO DE a_k

B : CALCULO DE b_k

1, 2, 3 y 4

Flags

Modo angular

Notacion

POTENCIACION DE UN DESARROLLO POLINOMICO

1.- Objeto.

Dado un desarrollo polinómico de la forma

$$P = 1 + a_1x + a_2x^2 + \dots$$

el siguiente programa obtiene los coeficientes del desarrollo

$$R = P^\alpha = 1 + b_1x + b_2x^2 + \dots$$

resultante de elevar P a la potencia α, siendo α real.

2.- Método.

Derivando en la igualdad $R = P^\alpha$ se obtiene:

$$R' = \alpha P^{\alpha-1} P', \text{ y por consiguiente } P R' - \alpha R P' = 0.$$

Teniendo en cuenta que:

$$P' = a_1 + 2a_2x + 3a_3x^2 + \dots + ia_ix^{i-1} + \dots$$

$$R' = b_1 + 2b_2x + 3b_3x^2 + \dots + ib_ix^{i-1} + \dots$$

y sustituyendo en la igualdad anterior:

$$(1 + a_1x + a_2x^2 + \dots)(b_1 + 2b_2x + \dots) - \alpha(1 + b_1x + \dots)(a_1 + 2a_2x + \dots) = 0$$

de la que se deduce:

$$b_1 - \alpha a_1 = 0 \quad \Rightarrow \quad b_1 = \alpha a_1$$

$$a_1b_1 + 2b_2 - \alpha(a_1b_1 + 2a_2) = 0 \quad \Rightarrow \quad b_2 = (a_1b_1(\alpha-1) + 2\alpha a_2)/2$$

Etc.

Se puede, pues, llegar a la siguiente fórmula de recurrencia:

$$b_i = \alpha a_i + \frac{1}{i} \sum_{j=1}^{i-1} a_{i-j} b_j \left[\alpha(i-j) - j\right]$$

que permite calcular todos los b_i.

3.- Observaciones.

Igual que en el programa anterior, la subrutina A que se enlista prevé el caso en que los a_i deban ser almacenados. En el paso 054 se ingresará el grado de P teniendo en cuenta que, al tener que ocupar un solo paso, deberá ser $N \leq 9$, siendo N dicho grado. Además, al haber disponibles 17 registros, también deberá ser $N + M \leq 17$, siendo M el número de términos de R deseados.

Si $a_i = f(i)$, se podrá programar bajo LBL A, no utilizándose lo enlistado y quedando disponibles los 17 registros para b_i, es decir, $M \leq 17$. Pero se deberá conservar el contenido de Y.

4.- Ejemplos.

1) Deseamos calcular los 10 primeros términos del desarrollo de $R = P^2$, siendo

$$P = e^x = 1 + x/1! + x^2/2! + x^3/3! + \ldots$$

con lo que $a_i = 1/i!$.

Sabemos que $R = P^2 = e^{2x} = 1 + 2x/1! + 4x^2/2! + 8x^3/3! + \ldots$

por lo que $b_i = 2^i/i!$, lo que nos servirá para comprobar la exactitud.

Programamos, pues

```
053   h LBL A
054   h  x!
055   h 1/x
056   h RTN
```

e iniciamos con 10; B .... 0,0003 (2 m 55 s).

Recuperando los b_i, formamos una tabla comparativa con 10 dígitos:

	Valor calculado	Valor exacto
RCL 1	2,000000000 E 0	2,000000000 E 0
RCL 2	2,000000000 E 0	2,000000000 E 0
RCL 3	1,333333333 E 0	1,333333333 E 0
RCL 4	6,666666668 E-1	6,666666667 E-1
RCL 5	2,666666667 E-1	2,666666667 E-1
RCL 6	8,888888883 E-2	8,888888889 E-2
RCL 7	2,539682547 E-2	2,539682540 E-2
RCL 8	6,349206304 E-3	6,349206349 E-3
RCL 9	1,410934761 E-3	1,410934744 E-3
RCL.0	2,821869439 E-4	2,821869489 E-4

2) Sea $P = 1 + 2x - 3x^2 + 4x^3$, deseando obtener $R = P^2$.

Aquí empleamos la subrutina A enlistada, con 3 ingresado en 054.

Hacemos, pues: 4; STO.7; 3; CHS; STO.6; 2; STO.5; 2; STO.9.

Y para 10 términos: 10; B .... 4,8667 E-8 (3 m 22 s)

RCL 1	4,0000	(b_1)
RCL 2	-2,0000	(b_2)
RCL 3	-4,0000	(b_3)
RCL 4	25,0000	(b_4)
RCL 5	-24,0000	(b_5)
RCL 6	16,0000	(b_6)
RCL 7	-1,0000 E-8	(b_7)
RCL 8	1,0000 E-8	(b_8)
RCL 9	-2,3333 E-8	(b_9)
RCL.0	4,8667 E-8	(b_{10})

A partir de b_6 debieran haber sido nulos, pero aparecen -- errores de redondeo.

Nota: Igual que en programa anterior, pag. 2.4.2.

HP-34C

TITULO: POTENCIACION DE UN DESARROLLO POLINOMICO.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA HASTA EL PASO 053			
2	SI SE HA DE EXPLICITAR a_k, CARGAR LOS PASOS RESTANTES, DESDE 054 AL FINAL, INGRESANDO POR [N_0] EL GRADO DEL POLINOMIO, Y CON $N \leq 9$			
2 bis	SI a_k ES CALCULABLE EN FUNCIÓN DE K, NO UTILIZAR DESDE 054 EN ADELANTE, PROGRAMANDO EN SU LUGAR LA OBTENCION DE a_k, SUPONIENDO A K EN EL REGISTRO X Y DEJANDO a_k EN EL MISMO REGISTRO. SE DEBERÁ CONSERVAR EL VALOR QUE INICIALMENTE OCUPE EL REGISTRO "Y" EN EL MISMO LUGAR.			
3	EN EL CASO DE a_k EXPLICITOS, ALMACENAR ESTOS HACIENDO	a_N	STO . 7	
		a_{N-1}	STO . 6	
		a_{N-2}	STO . 5	
		ETC.		
4	ALMACENAR EXPONENTE	α	STO . 9	
5	INGRESAR NÚMERO DE TÉRMINOS DEL DESARROLLO DESEADOS, CON $N+M \leq 17$, E INICIAR	M	B	b_M
6	RECUPERAR COEFICIENTES CALCULADOS HACIENDO, PARA $K=1,2,\dots,M$		RCL [K]	b_K
7	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: POTENCIACION DE UN DESARROLLO POLINOM.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL B	6	x	1		6		1		6	
2	EEX	7	RCL f I	2		7		2		7	
3	3	8	h INT	3		8		3		8	
4	÷	9	RCL 0	4		9		4		9	
5	1	040	−	5		110		5		180	
6	+	1	RCL .9	6		1		6		1	
7	STO f I	2	x	7		2		7		2	
8	h LBL 1	3	RCL 0	8		3		8		3	
9	RCL f I	4	−	9		4		9		4	
010	h INT	5	x	080		5		150		5	
1	STO 0	6	RCL f I	1		6		1		6	
2	GSB A	7	h INT	2		7		2		7	
3	RCL .9	8	÷	3		8		3		8	
4	x	9	RCL .8	4		9		4		9	
5	STO .8	050	+	5		120		5		190	
6	h LBL 2	1	STO .8	6		1		6		1	
7	RCL 0	2	GTO 2	7		2		7		2	
8	f x⇄I	3	h LBL A	8		3		8		3	
9	g DSE	4	[N_a]	9		4		9		4	
020	GTO 2	5	x⇄Y	090		5		160		5	
1	f x⇄I	6	f x>Y	1		6		1		6	
2	RCL .8	7	GTO 3	2		7		2		7	
3	STO f(i)	8	−	3		8		3		8	
4	g ISG	9	1	4		9		4		9	
5	GTO 1	060	7	5		130		5		200	
6	h RTN	1	−	6		1		6		1	
7	h LBL 2	2	f x⇄I	7		2		7		2	
8	RCL f(i)	3	RCL f(i)	8		3		8		3	
9	x⇄Y	4	x⇄Y	9		4		9		4	
030	ENTER↑	5	f x⇄I	100		5		170		5	
1	f x⇄I	6	g R↓	1		6		1		6	
2	STO 0	7	h RTN	2		7		2		7	
3	−	8	h LBL 3	3		8		3		8	
4	h INT	9	x	4		9		4		9	
5	GSB A	070	CLx	5		140		5		210	

Registros

0	J	1	b_1	2	b_2	3	b_3	4	Etc.	5		6	
7		8		9		.0		.1		.2		.3	
.4	Etc.	.5	a_{N_a-2}	.6	a_{N_a-1}	.7	a_{N_a}	.8	S	.9	α	I	I

Etiquetas

A: CALCULO O DIRECCIONAMIENTO DE a_k

B: CALCULO DE b_k

1, 2 y 3.

Flags

Modo angular

Notacion

EXPONENCIACION DE UN DESARROLLO POLINOMICO

1.- Objeto.

Dado un desarrollo polinomico de la forma

$$P = a_1x + a_2x^2 + a_3x^3 + \ldots$$

el siguiente programa obtiene los coeficientes del desarrollo

$$R = e^P = 1 + b_1x + b_2x^2 + b_3x^3 + \ldots$$

2.- Método.

Derivando la igualdad $R = e^P$ se obtiene

$$R' = P'e^P = P'R$$

Siendo: $P' = a_1 + 2a_2x + 3a_3x^2 + \ldots$

$R' = b_1 + 2b_2x + 3b_3x^2 + \ldots$

resultará

$$b_1 + 2b_2x + 3b_3x^2 + \ldots = (a_1 + 2a_2x + 3a_3x^2 + \ldots)(1 + b_1x + b_2x^2 + \ldots)$$

de la que se puede deducir:

$$\begin{array}{lcl} b_1 = a_1 & \Rightarrow & b_1 = a_1 \\ 2b_2 = 2a_2 + a_1b_1 & \Rightarrow & b_2 = a_2 + a_1b_1/2 \\ 3b_3 = 3a_3 + 2a_2b_1 + a_1b_1 & \Rightarrow & b_3 = a_3 + (2a_2b_1 + a_1b_2)/3 \end{array}$$

Etc.

que se puede generalizar en la siguiente fórmula recurrente:

$$b_i = a_i + \frac{1}{i}\sum_{j=1}^{i-1} ja_jb_{i-j} \quad , \qquad i = 1, 2, \ldots, m$$

en la que la suma será nula para $i = 1$, y m, el número de términos deseados.

3.- Observaciones.

Igual que en los dos programas anteriores, se ha enlistado una subrutina A para el caso de que a_i deban ser almacenados. Si -- N es el grado de P y M, el número de términos de R deseados, se deberá cumplir en este caso que $N + M \leqslant 18$.

Si $a_i = f(i)$ es programable, la subrutina A deberá efectuar su cálculo, y se podrá alcanzar $M \leqslant 18$.

4.- Ejemplos.

1) Sea $P = \ln(1 - x) = - x - x^2/2 - x^3/3 - \ldots - x^k/k$ y calculemos mediante el programa $R = e^P = 1 - x$. La respuesta correcta deberá ser $b_1 = -1$ y $b_i = 0$ para i 1.

Puesto que $a_i = -1/i$ programaremos la subrutina A

```
h  LBL A
h   1/x
    CHS
h   RTN
```

Los siete primeros términos calculados por el programa serían:

7; B	7,0000	(1 m 22 s)
RCL 1	-1,0000	
RCL 2	0,0000	
RCL 3	1,0000 E-10	
RCL 4	-2,5000 E-11	
RCL 5	-1,5000 E-11	
RCL 6	-1,0000 E-11	
RCL 7	-7,1429 E-12	

2) Dada la expresión $P = 2x + 3x^2$, calcular los 8 primeros términos de $R = e^P$.

Emplearemos, pues, la subrutina A enlistada, ingresando 2 en el paso 056. Después de almacenar los coeficientes haciendo

3; STO.8; 2; STO.7

iniciaremos:

8; B	8,0000	(2 m 13 s)
f FIX 9		
RCL 1	2,000000000	(b_1)
RCL 2	5,000000000	(b_2)
RCL 3	7,333333333	(b_3)
RCL 4	11,16666667	(b_4)
RCL 5	13,26666667	(b_5)
RCL 6	15,58888889	(b_6)
RCL 7	15,82539682	(b_7)
RCL 8	15,64801588	(b_8)

En realidad son pocos coeficientes para calcular R, pues producirían alto error por truncamiento.

Nota: Igual que la de la página 2.4.2.

HP-34C

TITULO: EXPONENCIACION DE UN DESARROLLO POLINOMICO.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA HASTA EL PASO 055			
2	SI SE HA DE EXPLICITAR a_K, CARGAR LOS PA-			
	SOS RESTANTES, DESDE 056 AL FINAL, INGRE-			
	SANDO POR N EL ORDEN DEL POLINOMIO.			
2 bis	SI a_K ES IMPLICITO Y CALCULABLE EN FUN-			
	CION DE k, NO UTILIZAR DESDE 056 AL FINAL,			
	PROGRAMANDO EN SU LUGAR LA OBTENCION DE			
	a_K, SUPONIENDO A k EN EL REGISTRO X Y DE-			
	JANDO a_K EN EL MISMO REGISTRO.			
3	EN EL CASO DE a_K EXPLICITOS, ALMACE-			
	NAR ESTOS HACIENDO	a_N	STO . 8	
		a_{N-1}	STO . 7	
		a_{N-2}	STO . 6	
		ETC.		
4	INGRESAR EL NUMERO DE TERMINOS DESEA-			
	DOS DEL DESARROLLO ($N+M \leq 18$) E INI-			
	CIAR	M	B	M
5	RECUPERAR TERMINOS HACIENDO, PARA k=1,			
	..., M		RCL k	b_K
6	PARA OTRO CASO, IR A 2.			

HP-34C PROGRAMA: EXPONENCIACION DE UN DESARROLLO POL.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL B	6	−	1		6		1		6	
2	EEX	7	GSB 3	2		7		2		7	
3	3	8	×	3		8		3		8	
4	÷	9	RCL f I	4		9		4		9	
5	1	040	h INT	5		110		5		180	
6	+	1	÷	6		1		6		1	
7	STO f I	2	RCL 0	7		2		7		2	
8	h LBL 1	3	×	8		3		8		3	
9	RCL f I	4	RCL .9	9		4		9		4	
010	h INT	5	+	080		5		150		5	
1	STO 0	6	STO .9	1		6		1		6	
2	GSB A	7	GTO 2	2		7		2		7	
3	STO .9	8	h LBL 3	3		8		3		8	
4	h LBL 2	9	f x⇄I	4		9		4		9	
5	RCL 0	050	RCL f(i)	5		120		5		190	
6	f x⇄I	1	x⇄y	6		1		6		1	
7	g DSE	2	f x⇄I	7		2		7		2	
8	GTO 2	3	g R↓	8		3		8		3	
9	f x⇄I	4	h RTN	9		4		9		4	
020	RCL .9	5	h LBL A	090		5		160		5	
1	STO f(i)	6	[N]	1		6		1		6	
2	g ISG	7	x⇄y	2		7		2		7	
3	GTO 1	8	f x>y	3		8		3		8	
4	RCL f I	9	GTO 4	4		9		4		9	
5	1	060	−	5		130		5		200	
6	−	1	1	6		1		6		1	
7	h INT	2	8	7		2		7		2	
8	h RTN	3	−	8		3		8		3	
9	h LBL 2	4	GTO 3	9		4		9		4	
030	f x⇄I	5	h LBL 4	100		5		170		5	
1	STO 0	6	×	1		6		1		6	
2	GSB A	7	CLx	2		7		2		7	
3	RCL f I	8	h RTN	3		8		3		8	
4	h INT	9		4		9		4		9	
5	RCL 0	070		5		140		5		210	

Registros

0	J	1	b_1	2	b_2	3	b_3	4	ETC.	5		6	
7		8		9		.0		.1		.2		.3	
.4		.5	ETC.	.6	a_{N-2}	.7	a_{N-1}	.8	a_N	.9	S	I	I

Etiquetas

A = CALCULO O DIRECCIONAMIENTO DE a_K

B = CALCULO DE b_K

1, 2, 3 y 4.

Flags

Modo angular

Notacion

ESTABILIDAD DE POLINOMIOS: CRITERIO DE ROUTH

1.- Objeto.

Dado un polinomio de grado $N \leqslant 17$, el siguiente programa determina si es estable, es decir, si todas sus raíces tienen la parte real negativa.

2.- Método.

Sea el polinomio $y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$ en el que los a_i son reales.

El criterio de Routh se aplica sobre el siguiente esquema:

$$
\begin{array}{llllcl}
a_n & a_{n-2} & a_{n-4} & \cdots & : & \\
a_{n-1} & a_{n-3} & a_{n-5} & \cdots & : & -a_n/a_{n-1} \\
b_{n-2} & b_{n-4} & b_{n-6} & \cdots & : & -a_{n-1}/b_{n-2} \\
c_{n-3} & c_{n-5} & c_{n-7} & \cdots & : & -b_{n-2}/c_{n-3} \\
d_{n-4} & d_{n-6} & d_{n-8} & \cdots & : & -c_{n-3}/d_{n-4} \\
\vdots & & & & & \vdots
\end{array}
$$

Las dos primeras filas contienen los coeficientes del polinomio y en su orden natural. La tercera fila se compone con las dos primeras mediante la siguiente operación: se multiplica la segunda fila por el factor $-a_n/a_{n-1}$, elegido de tal forma que, sumando luego la primera fila, se anule el primer coeficiente, mientras que los siguientes se anotarán como tercera fila, b_{n-2}, b_{n-4}, ..., desplazados un lugar a la izquierda:

$$b_i = a_i - a_{i-1}\, a_n/a_{n-1}$$

De igual forma se obtienen las sucesivas filas:

$$c_i = a_i - b_{i-1}\, a_{n-1}/b_{n-2}$$
$$d_i = b_i - c_{i-1}\, b_{n-2}/c_{n-3}$$

Etc.

Las raíces de la ecuación $y=0$ presentan sólo partes reales negativas cuando son positivos todos los números de la primera columna del esquema resultante, sin que sea nulo ninguno de ellos.

El llevar este esquema a un programa presenta evidente dificultad por el almacenamiento necesario. Sin embargo, observando que sólo se emplean las dos últimas filas para obtener la tercera y que los elementos se modifican alternadamente, puede disponerse de un arreglo unidimensional en el que cada fila "pise" a la penúltima. En definitiva, resultaría el siguiente algoritmo:

Para $i = n, (n-1), \ldots, 1$, hacer:

  Tomar $q = a_i/a_{i-1}$

  Para $j = (i-2), (i-4), \ldots$ hasta j 1, hacer

    $a_j = a_j - q\, a_{j-i}$

Finalizado el algoritmo, en lugar de los coeficientes iniciales, se tendrá almacenada la primera columna del esquema. Además, en caso de inestabilidad, el número de variaciones de signo observadas nos dará el número de raíces con parte real positiva.

3.- Ejemplo.

Se desea estudiar la estabilidad del polinomio

$$y = 6 + x + 4x^2 + x3 + 3x^4 + 9x^5 + 6x^6 + x^7.$$

Haremos: 6; STO 0; 1; STO 1; 4; STO 2; 1; STO 3; 3; STO 4; 9; STO 5; 6; STO 6; 1; STO 7.

Despueś de 7; A .... 7,0000 (39 s)

recuperamos los elementos de la primera columna:

RCL 0	6,0000
RCL 1	-291,8217
RCL 2	-0,2626
RCL 3	-11,9645
RCL 4	2,7647
RCL 5	8,5000
RCL 6	6,0000
RCL 7	1,0000

Luego es inestable. Además, produciéndose dos cambios de signo, se deduce que son dos las raices con parte real positiva. En efecto, resuelta la ecuación, las raíces son:

$-0,185681 \pm 0,861407\, i$

$0,658087 \pm 0,570197\, i$

$-1,531583 \pm 0,528902\, i$

$-3,881644$

con el segundo par conjugado con parte real positiva.

HP-34C

TITULO: ESTABILIDAD DE POLINOMIOS .- CRITERIO DE ROUTH.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR COEFICIENTES DEL POLINOMIO HA-			
	CIENDO, PARA $K=0,\dots,N$ $(N\leq 17)$	a_K	STO [K]	
3	INGRESAR GRADO E INICIAR	N	A	N
4	RECUPERAR NUEVOS COEFICIENTES HACIENDO			
	PARA $K=0,\dots,N$		RCL [K]	b_K
5	PARA UN NUEVO CASO IR A 2.			

HP-34C PROGRAMA : CRITERIO DE ROUTH.-

N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION
001	h LBL A	6	h LBL 3	1		6		1		6	
2	STO .9	7	1	2		7		2		7	
3	STO f I	8	−	3		8		3		8	
4	h LBL 1	9	f x⇄I	4		9		4		9	
5	RCL f(i)	040	RCL f(i)	5		110		5		180	
6	RCL f I	1	x⇄y	6		1		6		1	
7	STO .8	2	f x⇄I	7		2		7		2	
8	2	3	CLX	8		3		8		3	
9	EEX	4	+	9		4		9		4	
010	5	5	h RTN	080		5		150		5	
1	CHS	6		1		6		1		6	
2	+	7		2		7		2		7	
3	STO f I	8		3		8		3		8	
4	GSB 3	9		4		9		4		9	
5	÷	050		5		120		5		190	
6	ENTER↑	1		6		1		6		1	
7	ENTER↑	2		7		2		7		2	
8	ENTER↑	3		8		3		8		3	
9	h LBL 2	4		9		4		9		4	
020	g DSE	5		090		5		160		5	
1	GTO 2	6		1		6		1		6	
2	RCL .8	7		2		7		2		7	
3	STO f I	8		3		8		3		8	
4	g DSE	9		4		9		4		9	
5	GTO 1	060		5		130		5		200	
6	RCL .9	1		6		1		6		1	
7	h RTN	2		7		2		7		2	
8	h LBL 2	3		8		3		8		3	
9	RCL f I	4		9		4		9		4	
030	GSB 3	5		100		5		170		5	
1	x	6		1		6		1		6	
2	STO − (i)	7		2		7		2		7	
3	CLX	8		3		8		3		8	
4	+	9		4		9		4		9	
5	GTO 2	070		5		140		5		210	

Registros

0	a_0 , b_0	1	$a_1 . b_1$	2	$a_2 . b_2$	3	$a_3 . b_3$	4	a_4 , b_4	5	ETC.	6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6		.7		.8	J	.9	N	I	I

Etiquetas

A, 1, 2 y 3.

Flags

Modo angular

Notacion

EVALUACION DE POLINOMIOS DE CHEBYSHEV Y SUS RAICES

1.- Objeto.

El programa que sigue resuelve los siguientes casos:

1) Dado un polinomio ortogonal de Chebyshev, $T_n(x)$, y dado un valor x, $-1 \leq x \leq 1$, obtiene el valor numérico de T_n para el x dado, para cualquier n. (Sb. A).

2) Dada la expresión $y = d_oT_o + d_1T_1 + d_2T_2 + \ldots + d_nT_n$, en la que d_i son coeficientes reales y T_i polinomios de Chebyshev de grado i, calcula el valor numérico de y para un x dado, $-1 \leq x \leq 1$. En este caso se deberá cumplir $n \leq 16$. (Sub. B).

3) Calcula la raíces de $T_n(x) = 0$, de forma que

$$T_n(x) = \alpha_n(x - x_1)(x - x_2)\ldots(x - x_n)$$

en la que $\alpha_n = 2^{n-1}$, con $n \leq 37$. (Sb. 0).

2.- Método.

Los polinomios ortogonales de Chebyshev tienen como expresión trigonométrica: $T_n(x) = \cos(n\ t)$, $x = \cos t$. Sin embargo no se emplea esta expresión para su evaluación ya que, las rutinas de funciones trigonométricas, tienen una precisión limitada. Por ello se recurre mejor a las propiedades algebráicas de tales polinomios.

Estos, algebráicamente, son definibles según:

$$T_o = 1$$

$$T_1 = x$$

$$T_k = 2xT_{k-1} - T_{k-2}\ , \quad k = 2,\ 3,\ \ldots$$

La expresión general de T_k es una fórmula de recurrencia de tres términos que permite generar cualquier polinomio desde los dos primeros:

$$\begin{aligned} T_2 &= 2x^2 - 1 \\ T_3 &= 4x^3 - 3x \\ T_4 &= 8x^4 - 8x^2 + 1 \\ T_5 &= 16x^5 - 20x^3 + 5x \end{aligned}$$

Etc.

Sea ahora la expresión general

$$y = d_oT_o + d_1T_1 + d_2T_2 + \ldots + d_nT_n$$

Esta expresiòn es conocida como expansión o desarrollo de Chebyshev y tiene ciertas propiedades que la hacen muy util en problemas de aproximación algebráica de funciones. De aquí el interés de su evaluación.

Para ello, si en la expansión sustituímos T_n por su equivalente $T_n = 2xT_{n-1} - T_{n-2}$, en la que x es el valor de la variable a sustituir, el máximo grado de la expansión será ahora n-1. Continuando así hasta el final, mediante esta técnica de sustitución de adelante hacia atrás, conocida como multiplicación encajada de polinomios ortogonales, llegaremos al valor numérico de y con la menor pérdida de significación.

En definitiva, se puede establecer el siguiente algoritmo:

Tomar $T = d_n$

Si $n = 0$, parar.

En otro caso, hacer: $T_1 = T$; $T = d_{n-1} + A_{n-1}x\,T_1$

Si $n = 1$, parar.

En otro caso hacer, para $i = (n-2), (n-3), \ldots, 0$

$T_2 = T_1$; $T_1 = T$

$T = d_i + A_i x\,T_1 - T_2$

siendo siempre $A_0 = 1$ y $A_k = 2$ para $k > 0$. Al terminar será $y=p(x)= T$ el valor numérico de la expansión para el x dado.

Si lo que se quiere es evaluar un T_n determinado para un x dado, bastará, en el procedimiento anterior, hacer $d_n = 1$ y $d_i = 0$ para todo $i < n$, con lo que el mismo algoritumo es válido para ambos casos.

Por último, de la expresión trigonométrica de los polinomios se deduce el valor de las raíces que satisfacen $T_n = 0$. Tales raíces serán:

$$x_k = \cos \frac{2k-1}{2n}\pi \quad , k= 1, \ldots, n.$$

Las raíces se reparten en el eje de abscisas entre -1 y 1, con el origen como centro de simetría. Si el grado n es impar, una raiz será precisamente 0. Bastará, pues, calcular las de valor positivo, -- por ejemplo, Las restantes son del mismo valor absoluto pero con --- signo negativo.

3.- Observaciones.

El programa elaborado sigue el algoritmo expuesto. Mediante el "flag" 0, borrado o ajustado según se entre por A o B, se calcula $T_n(x)$ ó la expansión. Si el "flag" 0 está ajustado por la entrada B, el programa llamará a los registros que contienen a d_i. En este caso, el grado de la expansión está limitado por el número de registros disponibles, 16.

Si está borrado por la entrada A, no llamará a ningún registro y utilizará 1 ó 0 para d_i para i=n o no. Al no llamar a ningún registro se impide la detención por "Error 2" y el valor de n es ilimitado.

Ya que A_i es 1 para i= 0 ó 2 para los restantes, se emplea la prueba de 0 implícita en el paso 020 para determinar su valor y, además, para ajustar el "flag" 1 que detenga el algoritmo en la siguiente pasada.

Por último, el cálculo de las raices no ofrece dificultad y aplica la fórmula expuesta, obteniendo sólo las positivas, por lo que n puede llegar a ser 37.

4.- Ejemplos.

1) Calcular $T_5(0,3)$, $T_{10}(0,6)$ y $T_{21}(0,33)$.

Haremos:

a) 5; ENTER ↑ ; 0,3; A .... 0,998880 (12 s)

b) 10; ENTER ↑ ; 0,6; A ....-0,988497 (23 s)

c) 21; ENTER ↑ ; 0,33; A ...... 0,702703 (48 s)

2) Calcular el valor de

$$y = 3T_0 - 2T_1 - T_2 + 5T_3 + 4T_4 - 6T_5 + T_6$$

para x = 0,8.

Después de 3; STO 0; 2; CHS; STO 1; 1; CHS; STO 2; 5; STO 3; 4; STO 4; 6; CHS; STO 5; 1; STO 6; haremos

6; ENTER ↑ ; 0,8; B .... 1,217728 (16 s)

3) Calcular las raíces de T_8.

Haremos: 8; GSB 0 .... 128,000000 (α_8) (14 s)

RCL 1 0,980785
RCL 2 0,831470
RCL 3 0,555570
RCL 4 0,195090

Las cuatro restantes raíces tendrán el mismo valor absoluto con signo negativo.

Nota: El algoritmo de multiplicación encajada es una versión del de "Análisis Numérico". S.D.Conte y C.de Boor.2ªEd.McGraw-Hill.1974.

HP-34C

TITULO: EVALUACION DE POLINOMIOS DE CHEBYSHEV Y SUS RAICES.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PARA EVALUAR POLINOMIOS $T_n(x)$ PARA UN x DADO			
	HACER (n ILIMITADO)	n	ENTER↑	
		x	A	$T_n(x)$
3	PARA EVALUAR EXPANSIONES POLINOMICAS DE LA			
	FORMA $p(x) = d_0T_0 + d_1T_1 + \ldots + d_nT_n$ PARA UN			
	x DADO, HACER:			
3.1	ALMACENAR COEFICIENTES d_i HACIENDO. PARA			
	$i = 0, 1, \ldots, n$ $(n \leq 16)$	d_i	STO [i]	
3.2	INGRESAR VALORES E INICIAR	n	ENTER↑	
		x	B	$p(x)$
4	PARA OBTENER LAS RAICES DE $T_n(x) = 0$, PARA			
	$n \leq 37$, HACER	n	GSB 0	α
4.1	RECUPERAR LAS RAICES HACIENDO, PARA $k = 1, 2, \ldots,$			
	, INT $(n/2)$		RCL [k]	x_k
4.2	SOLO SE CALCULAN LAS RAICES POSITIVAS. LAS NE-			
	GATIVAS SON IGUALES CON SIGNOS MENOS. SI A-			
	DEMÁS n ES IMPAR, HAY UNA RAIZ CENTRAL CE-			
	RO. EL POLINOMIO QUEDA DESCOMPUESTO EN:			
	$T_n(x) = \alpha\,(x - x_1)(x - x_2)\ldots(x - x_n)$			
5	PARA OTRO CASO IR A 2, 3 ó 4 CORRESPONDIEN-			
	TE.			

HP-34C PROGRAMA : EVALUACION DE POLINOMIOS DE CHEBYSHEV.

N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION
001	h LBL A	6	h LBL 5	1		6		1		6	
2	h CF 0	7	RCL f(i)	2		7		2		7	
3	GTO 1	8	+	3		8		3		8	
4	h LBL B	9	GTO 2	4		9		4		9	
5	h SF 0	040	h LBL 3	5		110		5		180	
6	h LBL 1	1	2	6		1		6		1	
7	h CF 1	2	GTO 4	7		2		7		2	
8	STO .9	3	h LBL 0	8		3		8		3	
9	g R↓	4	g DEG	9		4		9		4	
010	STO f I	5	STO .9	080		5		150		5	
1	CLX	6	2	1		6		1		6	
2	STO .7	7	÷	2		7		2		7	
3	1	8	h INT	3		8		3		8	
4	h F? 0	9	STO f I	4		9		4		9	
5	RCL f(i)	050	h LBL 6	5		120		5		190	
6	h LBL 2	1	RCL f I	6		1		6		1	
7	h F? 1	2	2	7		2		7		2	
8	h RTN	3	×	8		3		8		3	
9	STO .8	4	1	9		4		9		4	
020	g DSE	5	−	090		5		160		5	
1	GTO 3	6	9	1		6		1		6	
2	h SF 1	7	0	2		7		2		7	
3	1	8	×	3		8		3		8	
4	h LBL 4	9	RCL .9	4		9		4		9	
5	×	060	÷	5		130		5		200	
6	RCL .9	1	f COS	6		1		6		1	
7	×	2	STO f(i)	7		2		7		2	
8	RCL .7	3	g DSE	8		3		8		3	
9	−	4	GTO 6	9		4		9		4	
030	RCL .8	5	2	100		5		170		5	
1	STO .7	6	RCL .9	1		6		1		6	
2	g R↓	7	1	2		7		2		7	
3	h F? 0	8	−	3		8		3		8	
4	GTO 5	9	h y^x	4		9		4		9	
5	GTO 2	070	STO 0	5		140		5		210	

Registros

0	d_0, α	1	d_1, x_1	2	d_2, x_2	3	d_3, x_3	4	d_4, x_4	5	d_5, x_5	6	d_6, x_6
7	d_7, x_7	8	d_8, x_8	9	d_9, x_9	.0	d_{10}, x_{10}	.1	d_{11}, x_{11}	.2	d_{12}, x_{12}	.3	d_{13}, x_{13}
.4	d_{14}, x_{14}	.5	d_{15}, x_{15}	.6	d_{16}, x_{16}	.7	T_{n-2}, x_{17}	.8	T_{n-1}, x_{18}	.9	x, n	I	i

Etiquetas

A : CALCULA $T_n = f(x)$.

B : CALCULA $p(x) = \sum_0^n d_i T_i$.

0 : CALCULA LAS RAICES DE $T_n(x) = 0$.

1, 2, 3, 4, 5 Y 6.

Flags

0 : DISCRIMINA ENTRE #A Y #B.

1 : FINAL DE ALGORITMO

Modo angular

DEG PARA #0.

Notacion

TRANSFORMACION DE UN POLINOMIO EN UNA ESPANSION DE CHEBYSHEV

1.- Objeto.

Dado un polinomio en la forma $p(x) = \sum_{i=0}^{n} a_i x^i$, en el que a_i son coeficientes reales, el siguiente programa lo transforma en $p(x) = \sum_{i=0}^{n} d_i T_i(x)$, en la que d_i son así mismo reales y $T_i(x)$, los polinomios ortogonales de Chebyshev de grado i. El programa permite $n \leqslant 15$.

La expresión de un polinomio en esta forma es utilizable en el acortamiento o economización de series de potencias. En general, con $-1 \leqslant x \leqslant 1$, se deberá cumplir $|T_i(x)| \leqslant 1$ para cualquier i. -- En consecuencia $|d_i T_i(x)| \leqslant |d_i|$. Por ello, la supresión de un término k supone un error acotado $\varepsilon \leqslant |d_i|$.

Si se suprimen varios términos se cumplirá

$$\varepsilon \leqslant |d_i| + |d_j| + |d_k| + \ldots$$

La condición $-1 \leqslant x \leqslant 1$ siempre se podrá conseguir mediante un cambio de variable en el polinomio original. (Ver 2.11).

2.- Método.

De la forma algebráica de los $T_i(x)$:

$$T_0 = 1 \qquad T_4 = 8x^4 - 8x^2 + 1$$
$$T_1 = x \qquad T_5 = 16x^5 - 20x3 + 5x$$
$$T_2 = 2x^2 - 1 \qquad T_6 = 32x^6 - 48x4 + 18x^2 - 1$$
$$T_3 = 4x^3 - 3x \qquad \text{Etc.}$$

se puede despejar cada una de las potencias de x de la siguiente manera:

$$x^0 = T_0 \qquad x^4 = 2^{-3}(3T_0 + 4T_2 + T_4)$$
$$x = T_1 \qquad x^5 = 2^{-4}(10T_1 + 5T_3 + T_5)$$
$$x^2 = 2^{-1}(T_0 + T_2) \qquad x^6 = 2^{-5}(10T_0 + 15T_2 + 6T_4 + T_5)$$
$$x^3 = 2^{-2}(3T_1 + T_3) \qquad \text{Etc.}$$

Multiplicadas estas expresiones por el a_i correspondiente y ordenando se llegaría a la expansión buscada.

La sistematización del método ha tenido ciertas dificultades al no encontrar bibliografía al respecto. Sin embargo se ha llegado al siguiente algoritmo que genera convenientemente los distintos coeficientes necesarios:

Hacer, para $i = 2, 3, \ldots, n$

$m = a_i/2^{i-1}$

$d_i = 0$

$k = i$

$L = 1$

Hacer, para $j = i, (i-2), (i-4), \ldots, j \geqslant 0$

Si $j \neq 0$, $d_j = d_j + m$

Si $j = 0$, $d_j = d_j + m/2$

$m = m\, k/L$

$k = k - 1$

$L = L + 1$

Terminado el algoritmo, los coeficientes d_i acuparán el lugar de los a_i.

3.- Ejemplo.

Como consecuencia de un ajuste, se ha llegado a la expresión $F = 21134{,}19 - 13217{,}79x - 2010{,}88x^2 + 3419{,}98x^3 + 729{,}18x^4 - 750{,}28x^5$, en la que x está en el intervalo$(-1,1)$.

Se pretende economizarla en lo posible, admitiéndose un error $\leqslant 500$.

Cargado el programa, hacemos:

21.134,19; STO 0; 13.217,79; CHS; STO 1; 2.010,88; CHS; STO 2; 3.419,98; STO 3; 729,18; STO 4; 750,28; CHS; STO 5.

5; A	5,0000	(42 s)
RCL 0	20.402,1925	
RCL 1	-11.121,7300	
RCL 2	-640,8500	
RCL 3	620,5325	
RCL 4	91,1475	
RCL 5	-46,8925	

Los únicos coeficientes cuyos valores absolutos suman un valor inferior a 500 son el 4º y el 5º. Luego la expresión acortada pasaría a ser:

$$F = 20.402{,}1925T_0 - 11.121{,}7300T_1 - 640{,}8500T_2 + 620{,}5325T_3$$

HP-34C

TITULO: TRANSFORMACION DE UN POLINOMIO EN UNA EXPANSION DE CHEBYSHEV.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR COEFICIENTES DEL POLINOMIO HA-			
	CIENDO, PARA $k = 0, \ldots, N$ $(N \leq 15)$	a_k	STO [k]	
3	INICIAR CALCULO INGRESANDO GRADO	N	A	N
4	RECUPERAR COEFICIENTES DE LA EXPANSION			
	HACIENDO. PARA $k = 0, \ldots, N$		RCL [k]	d_k
5	PARA OTRO CASO, IR A 2.			

HP-34C PROGRAMA : TRANSF. DE UN POLIN. EN UNA EXP. DE CHEBYSHEV.

N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION	N°	FUNCION
001	h LBL A	6	STO + (i)	1		6		1		6	
2	EEX	7	f R↑	2		7		2		7	
3	3	8	h LBL 3	3		8		3		8	
4	÷	9	2	4		9		4		9	
5	2	040	−	5		110		5		180	
6	+	1	STO . 9	6		1		6		1	
7	STO f I	2	×	7		2		7		2	
8	h LBL 1	3	g R↓	8		3		8		3	
9	RCL f (i)	4	STO f I	9		4		9		4	
010	g x=0	5	RCL . 6	080		5		150		5	
1	GTO 4	6	RCL . 8	1		6		1		6	
2	STO − (i)	7	×	2		7		2		7	
3	2	8	RCL . 7	3		8		3		8	
4	RCL f I	9	÷	4		9		4		9	
5	h INT	050	STO . 6	5		120		5		190	
6	STO . 8	1	RCL . 8	6		1		6		1	
7	STO . 9	2	1	7		2		7		2	
8	1	3	−	8		3		8		3	
9	STO . 7	4	STO . 8	9		4		9		4	
020	−	5	RCL . 7	090		5		160		5	
1	h y^x	6	1	1		6		1		6	
2	÷	7	+	2		7		2		7	
3	STO . 6	8	STO . 7	3		8		3		8	
4	h LBL 2	9	RCL . 9	4		9		4		9	
5	RCL . 6	060	g x<0	5		130		5		200	
6	2	1	GTO 4	6		1		6		1	
7	÷	2	GTO 2	7		2		7		2	
8	RCL . 9	3	h LBL 4	8		3		8		3	
9	f x⇄I	4	g ISG	9		4		9		4	
030	x⇄y	5	GTO 1	100		5		170		5	
1	STO + (i)	6	RCL f I	1		6		1		6	
2	RCL . 9	7	h INT	2		7		2		7	
3	g x=0	8	1	3		8		3		8	
4	GTO 3	9	−	4		9		4		9	
5	g R↓	070	h RTN	5		140		5		210	

Registros

0	a_0 , d_0	1	a_1 , d_1	2	a_2 , d_2	3	a_3 , d_3	4	ETC.	5		6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6	UTILIZ.	.7	l	.8	k	.9	j	I	i

Etiquetas

A : CALCULO

1, 2, 3 y 4.

Flags

Modo angular

Notacion

TRANSFORMACION DE UNA EXPANSION DE CHEBYSHEV EN UN POLINOMIO.-

FORMACION DE POLINOMIOS DE CHEBYSHEV.-

1.- Objeto.

Dada la expansión de Chebyshev

$$p(x) = d_oT_o + d_1T_1 + \dots + d_nT_n$$

en la que d_i son coeficientes reales y T_i, polinomios ortogonales de Chebyshev de grado i, el siguiente programa la transforma en

$$p(x) = a_o + a_1x + a_2x^2 + \dots + a_nx^n$$

es decir, la desarrolla en sumas de potencias de la variable, siendo a_i coeficientes reales y con $-1 \leq x \leq 1$. El programa admite $n \leq 15$.

Así mismo, calcula los coeficientes de la expresión algebráica de $T_n(x)$, para $n \leq 15$.

2.- Método.

Mediante la fórmula de recurrencia de tres términos, es posible construir un arreglo unidimensional en el que se vayan calculando los coeficientes de los sucesivos T_i para, así, hacer la sustitución en la expansión. Sin embargo, esta solución limitaría considerablemente n, ya que requeriría 2n registros más los de índice y auxiliares.

Por ello se ha recurrido a otro artificio. Sea la expresión de la 4ª potencia de x en función de T_i, por ejemplo:

$$x^4 = 2^{-3}(T_4 + 4T_2 + 3T_o)$$

de la que podemos deducir:

$$T_4 = 2^3x^4 - (4T_2 + 3T_o)$$

En general, se tendrá la expresión:

$$T_i = 2^{i-1}x^i - f(T_{i-2}, T_{i-4}, \dots)$$

en la que los coeficientes con los que participa T_{i-2}, T_{i-4}, etc., son calculables mediante el algoritmo del programa anterior (ver pag.2.9.1 y 2.9.2). Así, si en la expansión sustituimos T_n por su expresión anterior:

$$p(x) = d_oT_o + d_1T_1 + \dots + d_{n-2}T_{n-2} + d_{n-1}T_{n-1} + 2^{n-1}d_nx^n -$$
$$- d_n\ f(T_{n-2}, T_{n-4}, \dots)$$

el coeficiente del término en x^n será $a_n = 2^{n-1} d_n$, y ya no sufrirá variación por ser los otros términos de grado inferior. Por otra -- parte, el último término alterará los coeficientes de T_{n-2}, T_{n-4}, etc., llegándose a:

$$p(x) = d_o T_o + d_1 T_1 + \dots + d'_{n-2} T_{n-2} + d_{n-1} T_{n-1} + a_n x^n.$$

Si, sucesivamente, vamos sustituyendo T_{n-1}, T_{n-2}, etc. iremos obteniendo los sucesivos a_{n-1}, a_{n-2}, etc.

Lo anterior se puede expresar en el siguiente algoritmo:

```
Hacer, para i = n, (n-1), ..., 1
|   Tomar m = d_i; a_i = 2^(i-1) d_i; k = i; L = 1
|   Hacer, para j =(i-2), (i-4), ..., j ≥ 0
|   |   m = m k/L ;  L = L+1 ; k = k - 1
|   |   Si j ≠ 0, d_j = d_j - m
|___|___Si j = 0, d_j = d_j - m/2.
```

Al terminar el algoritmo, tendremos almacenados los a_i en lugar de los d_i. Esto permite mantener $n \leqslant 15$, como en el anterior programa.

Por último, si en la expansión hacemos $d_n = 1$ y los restantes $d_i = 0$, al aplicar el algoritmo anterior obtendremos los coeficientes de la forma algebráica de T_n.

3.- Ejemplo.

1) Sea la expansión $y = 20402,20 - 11121,73\ T_1 - 640,85\ T_2 + + 620,53\ T_3$

obtenida de un desarrollo mayor economizado (ver ejemplo del programa anterior, pag. 2.9.2). Para pasar a una expresión polinómica normal, haremos: 20.402,20; STO 0; 11.121,73; CHS; STO 1; 640,85; CHS; STO 2; 620,53; STO 3.

3; A	0,0000	(19 s)
RCL 0	21.043,0500	(a_o)
RCL 1	-12.983,3200	(a_1)
RCL 2	-1.281,7000	(a_2)
RCL 3	2.482,1200	(a_3)

2) Deseamos conocer los coeficientes de T_{11}.

Después de:	11; B	0,0000	(1 m 17 s)
	RCL 0	0,0000	
	RCL 1	-11,0000	
	Etc.		

$$T_{11}(x) = -11x + 220x^3 - 1232x^5 + 2816x^7 - 2816x^9 + 1024x^{11}$$

HP-34C

TITULO: TRANSFORMACION DE UNA EXPANSION DE CHEBYSHEV EN UN POLINOMIO.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA			
2	PARA TRANSFORMACION DE UNA EXPANSION, HACER:			
2.1	ALMACENAR COEFICIENTES HACIENDO, PARA k=0,1,			
	..., N ($N \leq 15$)	d_k	STO [k]	
2.2	INGRESAR GRADO E INICIAR	N	A	0,0000
2.3	RECUPERAR COEFICIENTES DEL POLINOMIO HA-			
	CIENDO, PARA k=0,1,...,N		RCL [k]	a_k
3	PARA OBTENER LOS COEFICIENTES DE T_N, CON			
	$N \leq 15$, HACER:			
3.1	INGRESAR GRADO E INICIAR	N	B	0,0000
3.2	OBTENER COEFICIENTES HACIENDO, PARA k=0,			
	..., N		RCL [k]	a_k
4	PARA OTROS CASOS IR A 3 ó 2.			

HP-34C PROGRAMA : TRANSF. DE UNA EXP. DE CHEBYSHEV EN UN POLINOM.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	g x<0	1		6		1		6	
2	STO f I	7	GTO 1	2		7		2		7	
3	h LBL 1	8	f x⇄I	3		8		3		8	
4	RCL f(i)	9	RCL .6	4		9		4		9	
5	STO .6	040	2	5		110		5		180	
6	g x=0	1	÷	6		1		6		1	
7	GTO 1	2	STO −(i)	7		2		7		2	
8	2	3	RCL .9	8		3		8		3	
9	RCL f I	4	g x=0	9		4		9		4	
010	STO .9	5	GTO 2	080		5		150		5	
1	STO .8	6	g R↓	1		6		1		6	
2	1	7	STO −(i)	2		7		2		7	
3	STO .7	8	f R↑	3		8		3		8	
4	−	9	h LBL 2	4		9		4		9	
5	h y^x	050	g R↓	5		120		5		190	
6	STO × (i)	1	g R↓	6		1		6		1	
7	h LBL 2	2	f x⇄I	7		2		7		2	
8	RCL .8	3	GTO 2	8		3		8		3	
9	RCL .6	4	h LBL 1	9		4		9		4	
020	×	5	g DSE	090		5		160		5	
1	RCL .7	6	GTO 1	1		6		1		6	
2	÷	7	RCL f I	2		7		2		7	
3	STO .6	8	h RTN	3		8		3		8	
4	RCL .7	9	h LBL B	4		9		4		9	
5	1	060	f REG	5		130		5		200	
6	+	1	STO f I	6		1		6		1	
7	STO .7	2	1	7		2		7		2	
8	RCL .8	3	STO f(i)	8		3		8		3	
9	1	4	GTO 1	9		4		9		4	
030	−	5		100		5		170		5	
1	STO .8	6		1		6		1		6	
2	RCL .9	7		2		7		2		7	
3	2	8		3		8		3		8	
4	−	9		4		9		4		9	
5	STO .9	070		5		140		5		210	

Registros

0	d_0 , a_0	1	d_1 , a_1	2	d_2 , a_2	3	d_3 , a_3	4	d_4 , a_4	5	ETC.	6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6	Utiliz.	.7	l	.8	k	.9	j	I	i

Etiquetas

1 y 2.

A: TRANSFORMACION DE UNA EXPANSION.

B: CALCULO COEFICIENTES DE T_n.

Flags

Modo angular

Notacion

CAMBIO DE VARIABLES EN POLINOMIOS

1.- Objeto.

1) Dado el polinomio $y = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$, en el que los a_i son reales y la variable x está definida $A \leq x \leq B$, el siguiente programa lo transforma en

$$y = b_0 + b_1u + b_2u^2 + \dots + b_nu^n$$

en el que la variable unificada u está definida $-1 \leq u \leq 1$, $n \leq 15$.

2) De igual forma, realiza la transformación contraria. Es decir, partiendo de $y = p(u)$, con $-1 \leq u \leq 1$, obtiene $y = p(x)$, con $A \leq x \leq B$, e, igual que antes, $n \leq 15$.

2.- Método.

El cambio de variable previsto supone:

$$x = \frac{B-A}{2}\left(u + \frac{B+A}{B-A}\right) = \alpha\,(u + \beta)$$

$$u = \frac{2}{B-A}\left(x - \frac{B+A}{2}\right) = \alpha\,(x + \beta)$$

Luego α y β varían según se trate de resolver el caso 1) o el 2), pero el método a aplicar será el mismo.

La sustitución se hace en dos fases. Primero se elimina α según el simple algoritmo:

Para $i = 0, 1, 2, \dots, n$, hacer

$$a_i' = a_i\,\alpha^i$$

Para la segunda fase, bastará aplicar el algoritmo de Horner con $z = \beta$ (ver programa "Algoritmo de Horner", pag. 2.2.1).

La utilidad de este programa es clara. Se hace necesario para la utilización de los dos programas anteriores: "Transformación de un polinomio en una expansión de Chebyshev", pag. 2.9.1, y el recíproco, "Transformación de una expansión de Chebyshev en un polinomio", pag. 2.10.1, en los que se requiere una u otra sustitución de variables antes o después de su empleo.

3.- Observaciones.

Con objeto de mantener $n \leq 15$, como en los programas antes citados, no se ha podido efectuar la sustitución sin interrupción.

La entrada A borra el "flag" 0 y calcula $\alpha = (B-A)/2$. La entrada B ajusta el "flag" 0 y calcula $\alpha = 2/(B-A)$. Después, ambas entradas efectúan el algoritmo antes expuesto, deteniéndose el programa.

Se hace necesario ahora reingresar A y B para calcular ρ según esté el "flag" 0. Si está borrado, $\rho = (B + A)/(B - A)$. Si está ajustado, $\rho = -(B + A)/2$. Se desarrolla después el algoritmo de Horner.

4.- Ejemplos.

1) Como consecuencia de un ajuste polinómico, se ha obtenido $F = 47842{,}25 - 94981{,}34x + 219967{,}84x^2 - 259397{,}28x^3 +$
$+ 131711{,}68x^4 - 24008{,}96x^5$.
en el que $0{,}5 \leq x \leq 1{,}5$.

Para su economización, se requiere expresarla en forma de desarrollo de Chebyshev, por lo que se hace necesario el cambio de variable $x \Rightarrow u$. Hacemos, pues:

47.842,25; STO 0; 94.981,34; CHS; STO 1; 219.967,84; STO 2; 259.397,28; CHS; STO 3; 131.711,68; STO 4; 24.008,96; CHS; STO 5.

0,5; ENTER↑ ; 1,5; ENTER↑ ; 5; A 0,0156
0,5; ENTER↑ ; 1,5; R/S 729,1800

El tiempo de ejecución total ha sido de 57 s. Se obtiene así:

RCL 0	21.134,19
RCL 1	-13.217,79
RCL 2	-2.010,88
RCL 3	3.419,98
RCL 4	729,18
RCL 5	-750,28

2) Después de economizada se ha obtenido:

$$F = 21043{,}05 - 12983{,}32u - 1281{,}70u^2 + 2482{,}12u^3$$

con $-1 \leq u \leq 1$, y en la que habrá que restituir el primitivo campo de la variable; es decir, $0{,}5 \leq x \leq 1{,}5$.

Después de almacenar los coeficientes (en realidad deberían de estar almacenados correctamente si se usó el programa anterior, 2.10): 0,5; ENTER↑ ; 1,5; ENTER↑ ; 3; B 16,0000
0,5; ENTER↑ ; 1,5; R/S .... -64.697,6800

RCL 0	22.025,93
RCL 1	43.857,84
RCL 2	-64.697,68
RCL 3	19.856,96

HP-34C

TITULO: CAMBIO DE VARIABLES EN POLINOMIOS.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR COEFICIENTES HACIENDO, PARA			
	$k=0,\dots,N$ $(N \leq 15)$	a_k	STO [k]	
3	PARA LA TRANSFORMACION $x \in \{A,B\}$ A			
	$u \in \{-1,1\}$ HACER:			
3.1	INGRESAR LIMITES: - INFERIOR	A	ENTER ↑	
	- SUPERIOR	B	ENTER ↑	
3.2	INGRESAR GRADO DEL POLINOMIO	N	A	α^{N+1}
3.3	IR A 5.			
4	PARA LA TRANSFORMACION $u \in \{-1,1\}$ A $x \in \{A,B\}$			
	HACER			
4.1	INGRESAR LIMITES - INFERIOR	A	ENTER ↑	
	- SUPERIOR	B	ENTER ↑	
4.2	INGRESAR GRADO	N	B	α^{N+1}
4.3	IR A 5.			
5	REINGRESAR LIMITES Y REANUDAR CALCULO	A	ENTER ↑	
		B	R/S	b_{N-1}
6	OBTENER NUEVOS COEFICIENTES HACIENDO, PARA			
	$k=0,\dots,N$		RCL [k]	b_k
7	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: CAMBIO DE VARIABLE EN POLINOMIOS.-

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	RCL .6	1	g DSE	6		1		6	
2	h CF 0	7	÷	2	GTO 7	7		2		7	
3	GTO 0	8	CHS	3	h x⇄(i)	8		3		8	
4	h LBL B	9	h LBL 2	4	RCL .8	9		4		9	
5	h SF 0	040	STO .6	5	f x⇄I	110		5		180	
6	h LBL 0	1	h LBL 3	6	g R↓	1		6		1	
7	STO .8	2	GSB 4	7	h x⇄(i)	2		7		2	
8	STO .7	3	RCL .7	8		3		8		3	
9	g R↓	4	1	9		4		9		4	
010	x⇄y	5	−	080		5		150		5	
1	−	6	STO .7	1		6		1		6	
2	2	7	g x≠0	2		7		2		7	
3	÷	8	GTO 3	3		8		3		8	
4	h F? 0	9	GTO 5	4		9		4		9	
5	h 1/x	050	h LBL 6	5		120		5		190	
6	STO .6	1	STO + (i)	6		1		6		1	
7	RCL .8	2	GTO 6	7		2		7		2	
8	EEX	3	h LBL 7	8		3		8		3	
9	3	4	h x⇄(i)	9		4		9		4	
020	÷	5	GTO 7	090		5		160		5	
1	STO f I	6	h LBL 4	1		6		1		6	
2	1	7	RCL .7	2		7		2		7	
3	h LBL 1	8	STO f I	3		8		3		8	
4	STO × (i)	9	h LBL 6	4		9		4		9	
5	RCL .6	060	RCL f (i)	5		130		5		200	
6	×	1	RCL .6	6		1		6		1	
7	g ISG	2	×	7		2		7		2	
8	GTO 1	3	g DSE	8		3		8		3	
9	R/S	4	GTO 6	9		4		9		4	
030	+	5	STO + 0	100		5		170		5	
1	2	6	h LBL 5	1		6		1		6	
2	÷	7	RCL .8	2		7		2		7	
3	CHS	8	STO f I	3		8		3		8	
4	h F? 0	9	RCL f (i)	4		9		4		9	
5	GTO 2	070	h LBL 7	5		140		5		210	

Registros

0	a_0 , b_0	1	a_1 , b_1	2	a_2 , b_2	3	a_3 , b_3	4	a_4 , b_4	5	a_5 , b_5	6	a_6 , b_6
7	a_7 , b_7	8	a_8 , b_8	9	a_9 , b_9	.0	a_{10} , b_{10}	.1	a_{11} , b_{11}	.2	a_{12} , b_{12}	.3	a_{13} , b_{13}
.4	a_{14} , b_{14}	.5	a_{15} , b_{15}	.6	α , β	.7	j	.8	N	.9		I	i

Etiquetas

A : EFECTÚA LA TRANSFORMACIÓN $x \in \{A,B\} \Rightarrow u \in \{-1,1\}$

B : EFECTÚA LA TRANSFORMACIÓN INVERSA : $u \in \{-1,1\} \Rightarrow x \in \{A,B\}$

0, 1, 2, 3, 4, 5, 6 y 7

Flags

0 : DISCRIMINA ENTRE AMBAS TRANSFORMACIONES.

Modo angular

Notacion

INTERPOLACION POLINOMICA

1.- Objeto.

Dado los $n+1$ puntos de apoyo, (x_o, y_o), (x_1, y_1), ..., --- (x_n, y_n), el siguiente programa obtiene los coeficientes del polinomio real: $y = a_o + a_1x + a_2x^2 \ldots + a_nx^n$, que pasa por aquellos, - para $n \leqslant 7$, y con $x_i \neq x_j$ para cualesquiera i y j.

2.- Método.

Conocidos los $n+1$ puntos, se obtienen los cocientes de diferencias según:

$$(x_ox_1) = \frac{y_1 - y_o}{x_1 - y_o} \; ; \quad (x_1x_2) = \frac{y_2 - y_1}{x_2 - x_1} \; ; \ldots ; \; (x_{n-1}x_n) = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}$$

$$(x_ox_1x_2) = \frac{(x_1x_2)-(x_ox_1)}{x_2 - x_o} \; ; \quad (x_1x_2x_3) = \frac{(x_2x_3)-(x_1x_2)}{x_3 - x_1} \; ; \ldots$$

y por fin:

$$(x_ox_1\ldots x_n) = \frac{(x_1x_2\ldots x_n)-(x_ox_1\ldots x_{n-1})}{x_n - x_o}$$

El polinomio interpolante, dado con los coeficientes de Newton, adquiere la forma:

$$y = \lambda_o + \lambda_1(x-x_o) + \lambda_2(x-x_o)(x-x_1) + \ldots + \lambda_n(x-x_o)\ldots(x-x_{n-1})$$

en el que:

$$\lambda_o = y_o \; ; \quad \lambda_1 = (x_ox_1) \; ; \quad \lambda_2 = (x_ox_1x_2) \; ; \; \ldots \; ; \quad \lambda_n = (x_ox_1\ldots x_n)$$

Para obtener la notación normal se recurre al algoritmo de Horner modificado:

Dados los $n+1$ coeficientes $\lambda_o, \lambda_1, \ldots, \lambda_n$ para la forma de Newton, basados en los puntos $x_o, x_1, \ldots, x_{n-1}$ del polinomio, dado ademas el número z:

Tomar $a_n = \lambda_n$
Para i =(n-1), (n-2), ..., 0, hacer
  Tomar $a_i = \lambda_i + (z-x_i)a_{i+1}$

Entonces, $a_o, a_1, \ldots, a_n$ son los coeficientes de Newton basados en los puntos $z, x_1, x_2, \ldots, x_{n-2}$. Repitiendo el algoritmo n veces y para z=0, obtenemos la forma normal definitiva.

3.- Observaciones.

Las sucesivas diferencias divididas se disponen en un arreglo unidimensional, ya que sólo hay que retener las que constituyen los coeficientes de Newton. Así, cuando termina el cálculo, los coeficientes λ_o, λ_1, ..., λ_n quedan almacenados en lugar de las ordenadas y_o, y_1, ..., y_n. Las abscisas no se pierden ni sufren alteración.

Se ha previsto la entrada B para el cálculo del polinomio resultante para cualquier x. Además se puede emplear para comprobar la interpolación: entrando a B con uno de los x_i utilizados, nos deberá dar el y_i correspondiente. Para ello también se ha previsto la entrada O que extrae de la memoria el x_i para un i dado.

Por las necesidades de almacenamiento sólo se ha podido -- alcanzar $n \leq 7$. Sin embargo es en la práctica suficiente ya que, con valores muy altos de n se suelen formar polinomios inestables.

Por supuesto se deberá cumplir $x_i \neq x_j$ cualesquieran sean i y j.

4.- Ejemplo.

Sean los puntos

x	-3	-2	-1	0	1	2	3	4
y	35143	1415	15	1	-1	-597	-18905	-240475
i	0	1	2	3	4	5	6	7

Ingresados en el orden i, los y en los registros R_o, R_1, ..., R_7, y los x en los $R_{.o}$, $R_{.1}$, ..., $R_{.7}$, haremos:

7; A .... 8,0000 (tras 4 m 4 s)

Para comprobar el resultado haremos:

0; GSB 0 .... -3,0000 (x_o)
B ....35.143,0000 (y_o)

1; GSB 0 .... -2,0000 (x_1)
B .... 1.415,0000 (y_1)

Etc. Resultando correcta.

Los coeficientes serán:

RCL 0 .... 1,0000 (a_o)
RCL 1 .... 433,0000 (a_1)
RCL 2 .... 38,0000 (a_2)
RCL 3 .... -626,0000 (a_3)
RCL 4 .... -48,0000 (a_4)
RCL 5 .... 214,0000 (a_5)
RCL 6 .... 16,0000 (a_6)
RCL 7 .... -29,0000 (a_7)

Si se desea calcur: $y = \bar{p}(2,5)$: 2,5; B .... -3.230,7578

HP-34C

TITULO: INTERPOLACION POLINOMICA

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA:			
2	ALMACENAR LOS N+1 PUNTOS DE APOYO (N≤7):			
2.1	ALMACENAR ORDENADAS HACIENDO, PARA $k=0,\ldots,N$	Y_k	STO [k]	
2.2	ALMACENAR ABSCISAS HACIENDO, PARA $k=0,\ldots,N$	X_k	STO . [k]	
3	INGRESAR GRADO E INICIAR	N	A	N+1
4	PARA COMPROBAR INTERPOLACION HACER, PARA CUALESQUIERA k ENTRE 0 Y N:	k	GSB 0	X_k
			B	Y_k
5	OBTENER COEFICIENTES DEL POLINOMIO INTERPOLANTE HACIENDO, PARA $k=0,\ldots,N$		RCL [k]	a_k
6	PARA EVALUAR EL POLINOMIO PARA CUALQUIER ABSCISA X, HACER	X	B	Y
7	PARA OTRO CASO, IR A 2.			

HP-34C PROGRAMA : INTERPOLACION POLINOMICA.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	STO + 8	1	RCL 8	6		1		6	
2	STO 9	7	RCL 9	2	−	7		2		7	
3	h CF 0	8	RCL 8	3	h LBL 0	8		3		8	
4	h LBL 1	9	f x≤y	4	1	9		4		9	
5	1	040	GTO 2	5	0	110		5		180	
6	STO 8	1	h F? 0	6	+	1		6		1	
7	h LBL 2	2	h RTN	7	h LBL 8	2		7		2	
8	RCL 8	3	h SF 0	8	f x⇄I	3		8		3	
9	1	4	GTO 1	9	RCL f(i)	4		9		4	
010	− .	5	h LBL 4	080	x⇄y	5		150		5	
1	EEX	6	GSB 7	1	f x⇄I	6		1		6	
2	3	7	RCL f(i)	2	g R↓	7		2		7	
3	÷	8	×	3	h RTN	8		3		8	
4	RCL 9	9	GTO 5	4		9		4		9	
5	+	050	h LBL 3	5		120		5		190	
6	STO f I	1	h F? 0	6		1		6		1	
7	h LBL 3	2	STO −(i)	7		2		7		2	
8	h F? 0	3	GTO 3	8		3		8		3	
9	GTO 4	4	h LBL B	9		4		9		4	
020	RCL f I	5	STO 8	090		5		160		5	
1	1	6	RCL 9	1		6		1		6	
2	−	7	STO f I	2		7		2		7	
3	GSB 8	8	CLX	3		8		3		8	
4	STO −(i)	9	h LBL 6	4		9		4		9	
5	RCL f I	060	RCL f(i)	5		130		5		200	
6	GSB 0	1	+	6		1		6		1	
7	GSB 7	2	RCL 8	7		2		7		2	
8	−	3	×	8		3		8		3	
9	STO ÷(i)	4	g DSE	9		4		9		4	
030	h LBL 5	5	GTO 6	100		5		170		5	
1	g DSE	6	RCL 0	1		6		1		6	
2	GTO 3	7	+	2		7		2		7	
3	h F? 0	8	h RTN	3		8		3		8	
4	STO −(i)	9	h LBL 7	4		9		4		9	
5	1	070	RCL f I	5		140		5		210	

Registros

0	y_0 , a_0	1	y_1 , a_1	2	y_2 , a_2	3	y_3 , a_3	4	y_4 , a_4	5	y_5 , a_5	6	y_6 , a_6
7	y_7 , a_7	8	j , x	9	N	.0	x_0	.1	x_1	.2	x_2	.3	x_3
.4	x_4	.5	x_5	.6	x_6	.7	x_7	.8		.9		I	i

Etiquetas

A : CALCULA POLINOMIO INTERPOLANTE.

B : CALCULA y = P(x) PARA X DADO.

D : DIRECCIONAMIENTO DE x_k

1, 2, 3, 4, 5, 6, 7 y 8

Flags

0 : DISCRIMINA LOS DOS CICLOS NECESARIOS.

Modo angular

Notacion

DESARROLLO DE FOURIER DE VALORES DE FUNCION DISCRETOS.-

1.- Objeto.

Dados los m puntos y_o, y_1, ..., y_{m-1}, correspondientes a m ordenadas igualmente espaciadas en el intervalo 0, 2π, y supuesto que tales valores se repiten en los sucesivos intervalos de extensión 2 , es decir, que las ordenadas y_i corresponden a una función periódica, el siguiente programa obtiene el desarrollo de Fourier o aproximación trigonométrica a tales valores dada por

$$y = a_o + a_1 \cos x + a_2 \cos 2x + \ldots + a_n \cos nx + \\ + b_1 \operatorname{sen} x + b_2 \operatorname{sen} 2x + \ldots + b_n \operatorname{sen} nx$$

en la que $n \leq 8$ y m puede tener cualquier valor.

Respecto a las ordenadas, y_o debe corresponder al origen 0 y las siguientes a los subintervalos de extensión $2\pi/m$, de tal forma que la correspondiente a 2π deberá cumplir $y_m = y_o$.

Siendo el máximo orden n = 8, se cumple:

- Si $m \leq 16$ el desarrollo llega a n= INT(m/2) y es una interpolación exacta a los m puntos, equivalente, pues, a un polinomio de grado m-1.
- Si $m > 16$, el desarrollo llega n= 8 y será un ajuste por mínimos cuadrados a los m puntos.

2.- Método.

Si la función a desarrollar es f(x), el desarrollo anterior debe tener la mejor coincidencia. Aplicando el método de mínimos cuadrados:

$$Q = \int_0^{2\pi} (y - f(x))^2 \, dx = \text{mín.}$$

Desarrollando las propiedades del mínimo y por la ortogonalidad de las funciones trigonométricas se llega a

$$a_o = \frac{1}{2\pi} \int_0^{2\pi} f(x) \, dx \qquad a_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos nx \, dx$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(x) \operatorname{sen} nx \, dx \qquad n = 1, 2, \ldots$$

Estos coeficientes de Fourier de la función periódica f(x) resultan independientes del índice de los armónicos que se consideran, de forma que, si se prosigue el orden del desarrollo, los coeficientes ya calculados permanecen invariables.

En el cálculo numérico se puede plantear desde un principio como sumas finitas, ya que la ortogonalidad de las funciones -- trigonométricas, excepcionalmente, persiste cuando se trata de sumas. Partiendo, pues, de :

$$Q = \sum_{j=0}^{M-1} (y - f(x))^2 = \text{mín.}$$

se llegaría a

$$a_o = \frac{1}{M} \sum_{j=0}^{M-1} y_j \qquad a_i = \frac{2}{M} \sum_{j=0}^{M-1} y_j \cos ix_j$$

$$a_{M/2} = \frac{1}{M} \sum_{j=0}^{M-1} (-1)^j y_j \qquad b_i = \frac{2}{M} \sum_{j=0}^{M-1} y_j \text{ sen } ix_j$$

para $i = 1, 2, \ldots, M/2 - 1$

Tal como se plantea en este caso, las abscisas x_j se deducen facilmente ya que: $x_j = 2\pi j/M$. Entonces basta conocer el orden de la ordenada para saber la abscisa. Aquellas habrá que ingresarlas en el orden correspondiente una a una.

3.- Ejemplo.

Como ejemplo se ha escogido una sucesión de valores que corresponden a las amplitudes de una onda senoidal idealmente rectificada cada 30º. La función es así y = ABS(sen x) y los valores a ingresar serán:

k	y_k
0	0,0000
1	0,5000
2	0,8660
3	1,0000
Etc.	

Siguiendo las instrucciones, haríamos:

```
12; A  ................  0,0000
0; R/S ................  1,0000
0,5; R/S ..............  2,0000
...
0,8660; R/S ........... 11,0000
0,5; R/S ..............  6,0000     (N)
```

Al recuperar los coeficientes obtenemos:

```
RCL 0 ........ 0,622000
RCL 1 ........ 1,900000 E-10      RCL.1 ...... 1,300000 E-10
RCL 2 ........-0,455333           RCL.2 ...... 3,000000 E-11
RCL 3 ........ 9,379167 E-10      RCL.3 ...... 4,200000 E-10
RCL 4 ........-0,122000           RCL.4 ......-1,250000 E-9
RCL 5 ........ 1,280000 E-9       RCL.5 ......-6,500000 E-10
RCL 6 ........-0,089333           RCL.6 ......-6,870933 E-10
```

Si se hubiese integrado la función verdadera hubiese resultado $b_i = 0$ y $a_i = 0$ para i impar.

HP-34C

TITULO: DESARROLLO DE FOURIER DE VALORES DE FUNCION DISCRETOS.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	INGRESAR NUMERO DE PUNTOS DISPONIBLES IGUALMENTE ESPACIADOS EN EL PERIODO 2π E INICIAR	M	A	0,0000
3	PARA $k = 0, \ldots, M-1$. INGRESAR LOS VALORES DE LA FUNCION A APROXIMAR, HACIENDO	Y_k	R/S	$k+1$
4	TRAS INGRESAR EL ULTIMO VALOR Y_{M-1}, MUESTRA EL ORDEN DE LA APROXIMACION	Y_{M-1}	R/S	N
	(SI $M > 16$, $N = 8$ Y EL DESARROLLO ES UN AJUSTE POR MINIMOS CUADRADOS. SI $M \leq 16$, $N = INT(M/2)$ Y EL DESARROLLO ES UNA INTERPOLACION EXACTA A LOS M VALORES).			
5	OBTENER COEFICIENTES DEL DESARROLLO HACIENDO. PARA $k = 0, \ldots, N$		RCL [k]	a_k
	PARA $k = 1, \ldots, N$		RCL . [k]	b_k
6	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: DESARROLLO DE FOURIER: VALORES DISCRE.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	CLx	1		6		1		6	
2	f REG	7	RCL . 9	2		7		2		7	
3	STO . 0	8	RCL f I	3		8		3		8	
4	g RAD	9	h INT	4		9		4		9	
5	h LBL 1	040	×	5		110		5		180	
6	RCL 9	1	x⇄y	6		1		6		1	
7	R/S	2	f →R	7		2		7		2	
8	RCL . 0	3	STO + (i)	8		3		8		3	
9	÷	4	g R↓	9		4		9		4	
010	STO + 0	5	RCL f I	080		5		150		5	
1	2	6	1	1		6		1		6	
2	×	7	0	2		7		2		7	
3	x⇄y	8	+	3		8		3		8	
4	2	9	f x⇄I	4		9		4		9	
5	h π	050	x⇄y	5		120		5		190	
6	×	1	STO + (i)	6		1		6		1	
7	×	2	x⇄y	7		2		7		2	
8	RCL . 0	3	f x⇄I	8		3		8		3	
9	÷	4	×	9		4		9		4	
020	STO . 9	5	g ISG	090		5		160		5	
1	CLx	6	GTO 2	1		6		1		6	
2	RCL . 0	7	1	2		7		2		7	
3	2	8	STO + 9	3		8		3		8	
4	÷	9	RCL 9	4		9		4		9	
5	h INT	060	RCL . 0	5		130		5		200	
6	8	1	f x>y	6		1		6		1	
7	f x≤y	2	GTO 1	7		2		7		2	
8	x⇄y	3	RCL f I	8		3		8		3	
9	CLx	4	h INT	9		4		9		4	
030	EEX	5	1	100		5		170		5	
1	3	6	−	1		6		1		6	
2	÷	7	h RTN	2		7		2		7	
3	STO f I	8		3		8		3		8	
4	g ISG	9		4		9		4		9	
5	h LBL 2	070		5		140		5		210	

Registros

0	a_0	1	a_1	2	a_2	3	a_3	4	a_4	5	a_5	6	a_6
7	a_7	8	a_8	9	i	.0	M	.1	b_1	.2	b_2	.3	b_3
.4	b_4	.5	b_5	.6	b_6	.7	b_7	.8	b_8	.9	x	I	j

Etiquetas

A, 1 y 2.

Flags

Modo angular

RAD

Notacion

DESARROLLO DE FOURIER DE FUNCIONES DISCRETIZADAS

1.- Objeto.

Dada la función contínua f(x) para $0 \leq x \leq 2\pi$, el siguiente programa calcula los coeficientes del desarrollo de Fourier

$$f(x) = a_o + a_1 \cos(x) + a_2 \cos(2x) + \ldots + a_n \cos(nx) + \\ + b_1 \operatorname{sen}(x) + b_2 \operatorname{sen}(2x) + \ldots + b_n \operatorname{sen}(nx)$$

con $n \leq 7$. La función f(x) que se desea aproximar deberá ser programable, debiéndose tener en cuenta que, si para algun valor de x = z presentase su derivada alguna discontinuidad, es decir, si f(x) presentara algún salto de forma que para $f(z - 0) = z_1$ y para --- $f(z + 0) = z_2$, el programa que calcule f(x) deberá tener en cuenta tal discontinuidad tomando como valor de la función el valor medio, es decir $f(z) = (z_1 + z_2)/2$.

2.- Método.

El método seguido es idéntico al del programa anterior, o - sea, se calculan los coeficientes de acuerdo con

$$a_o = \frac{1}{M}\sum_{j=0}^{M-1} f(x_j) \quad ; \quad a_i = \frac{2}{M}\sum_{j=0}^{M-1} f(x_j)\cos(ix_j) \quad ; \quad b_i = \frac{2}{M}\sum_{j=0}^{M-1} f(x_j)\operatorname{sen}(ix_j)$$

para $i = 1, 2, \ldots, M/2 - 1$, y $x_j = 2\pi j/M$.

En esta ocasión M es definida por el usuario teniendo en cuenta la precisión deseada. A mayor M mayor exactitud en la integración de los coeficientes. Si $M > 14$, entonces n=7 y el desarrollo es un ajuste por mínimos cuadrados. Si $M \leq 14$, n=M/2 y el desarrollo es una interpolación exacta a los M puntos con $b_n=0$. Puesto que, a diferencia del programa anterior, hay libertad para la escogencia de M, se preceptúa que ésta sea par.

3.- Ejemplo.

Sea el mismo ejemplo de antes, y= ABS(sen x), perteneciente a una onda idealmente rectificada. Bajo B la programamos haciendo

```
059   LBL B
060   f SIN
061   h ABS
062   h RTN
```

Se han efectuados dos casos para distintos valores de M. Una vez corridos los dos ejemplos, se tabulan a continuación los resultados comparándolos con los valores correctos:

	M=12 (4 m 18 s)	M=24 (10 m 10 s)	Val. exactos
a_0	0,622008	0,632980	0,636620
a_1	1,300000 E-10	-3,600000 E-10	0
a_2	-0,455342	-0,431795	-0,424413
a_3	-8,620833 E-10	.5,400000 E-10	0
a_4	-0,122008	-0,092582	-0,084883
a_5	9,300000 E-10	8,590000 E-10	0
a_6	-0,089316	-0,044658	-0,036378
a_7	1,[illegible]000 E-9	1,242000 E-9	0

Los valores de b_i, que deben ser todos nulos, han resultado con órdenes E-9 y E-10. Como se aprecia, para conseguir algo de exactitud, se requiere valores de M elevados lo que resulta en altos tiempos de ejecución. Esto se agudiza tanto más cuantas mayores discontinuidades tenga la función en su derivada, como pasa en la del ejemplo en los puntos 0 y π.

HP-34C

TITULO: DESARROLLO DE FOURIER DE FUNCIONES DISCRETIZADAS.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PROGRAMAR BAJO LBL B EL CALCULO DE Y=F(X) SUPONIENDO A X EN EL REGISTRO X Y DEJANDO A Y EN EL MISMO REGISTRO. HAY DISPONIBLES DEL PASO 060 AL 084 CON NINGUN REGISTRO Ó DEL PASO 060 AL 070 CON R.8 Y R.9. TAMBIEN SE DISPONE DE LBL 1 A LBL 9.			
3	PREPARAR PROGRAMA HACIENDO		f REG	
			g RAD	
4	ALMACENAR NUMERO DE PUNTOS DE APOYO POR	M	STO . 0	
5	INICIAR		A	M
6	RECUPERAR COEFICIENTES DEL DESARROLLO			
	HACIENDO, PARA $k=0,\dots,N$		RCL [k]	a_k
	PARA $k=1,\dots,N$		RCL . [k]	b_k
	(SI $M>14$, $N=7$ Y EL DESARROLLO ES UN AJUSTE POR MINIMOS CUADRADOS. SI $M \leq 14$, $N=M/2$, Y EL DESARROLLO ES UNA INTERPOLACION EXACTA A LOS M PUNTOS, CON $b_N=0$).			
7	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: DESARROLLO DE FOURIER DE FUNC. DISCRETIZ.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	x⇄y	1		6		1		6	
2	RCL 9	7	f →R	2		7		2		7	
3	h π	8	STO + (i)	3		8		3		8	
4	×	9	g R↓	4		9		4		9	
5	2	040	RCL f I	5		110		5		180	
6	×	1	1	6		1		6		1	
7	RCL .0	2	0	7		2		7		2	
8	÷	3	+	8		3		8		3	
9	STO 8	4	f x⇄I	9		4		9		4	
010	GSB B	5	x⇄y	080		5		150		5	
1	RCL .0	6	STO + (i)	1		6		1		6	
2	÷	7	x⇄y	2		7		2		7	
3	STO + 0	8	f x⇄I	3		8		3		8	
4	2	9	×	4		9		4		9	
5	×	050	g ISG	5		120		5		190	
6	ENTER↑	1	GTO 0	6		1		6		1	
7	ENTER↑	2	1	7		2		7		2	
8	RCL .0	3	STO + 9	8		3		8		3	
9	2	4	RCL 9	9		4		9		4	
020	÷	5	RCL .0	090		5		160		5	
1	7	6	f x>y	1		6		1		6	
2	f x≤y	7	GTO A	2		7		2		7	
3	x⇄y	8	h RTN	3		8		3		8	
4	CLX	9	h LBL B	4		9		4		9	
5	EEX	060		5		130		5		200	
6	3	1		6		1		6		1	
7	÷	2		7		2		7		2	
8	STO f I	3		8		3		8		3	
9	g ISG	4		9		4		9		4	
030	h LBL 0	5		100		5		170		5	
1	CLX	6		1		6		1		6	
2	RCL 8	7		2		7		2		7	
3	RCL f I	8		3		8		3		8	
4	h INT	9		4		9		4		9	
5	×	070		5		140		5		210	

Registros

0	a_0	1	a_1	2	a_2	3	a_3	4	a_4	5	a_5	6	a_6
7	a_7	8	x	9	i	.0	M	.1	b_1	.2	b_2	.3	b_3
.4	b_4	.5	b_5	.6	b_6	.7	b_7	.8		.9		I	j

Etiquetas

A y 0.

B : CALCULA Y = F(X)

Flags

Modo angular

RAD (MANUAL)

Notacion

DESARROLLO DE FOURIER POR INTEGRACION DE COEFICIENTES

1.- Objeto.

Dada la función contínua $f(x)$ para $0 \leqslant x \leqslant 2\pi$, el siguiente programa calcula los coeficientes de Fourier, uno a uno, del desarrollo

$$f(x) = a_0 + a_1 \cos(x) + a_2 \cos(2x) + \ldots + a_n \cos(nx) + b_1 \operatorname{sen}(x) + b_2 \operatorname{sen}(2x) + \ldots + b_n \operatorname{sen}(nx)$$

para cualquier n. La función $f(x)$ deberá ser programable, debiéndose tener en cuenta que, si para algún valor de $x = z$ presentase su derivada alguna discontinuidad, es decir, si $f(x)$ presentara algún salto de forma que para $f(z-0)=z_1$ y para $f(z+0)=z_2$, el programa que calcule $f(x)$ deberá tener en cuenta tal discontinuidad tomando como valor de la función el valor medio, es decir, $f(z)=(z_1+z_2)/2$.

Además, y para evitar el cálculo de coeficientes que deberán ser nulos, se tendrán en cuenta las siguientes reglas:

-Si la función $f(x)$ es de simetría par, es decir, es simétrica respecto al eje y, cumpliéndose por tanto $f(-x)=f(x)$, se deberá cumplir $b_i = 0$ para cualquier i.

-Si la función es de simetría impar, es decir, simétrica respecto al origen, cumpliéndose por tanto $f(-x)=-f(x)$, se deberá cumplir $a_i=0$ para cualquier i.

-Si la función es de simetría completa, llamada así una combinación de las anteriores que podría expresarse como $f(-x) = \pm f(x)$, y que cumple la propiedad de que $f(x+\pi)=-f(x)$, por la cual se repite la forma de la curva en cada cuarto de período, entonces deberá cumplirse $a_0 = a_i = b_i = 0$ para i= 2, 4, 6, etc.

2.- Método.

No es facil resistir la tentación de calcular los coeficientes de Fourier por integración directa empleando la potente función $\int_x^y$ de que está provista la HP-34C. Sin embargo se prevén largas duraciones de los cálculos de coeficientes de órdenes altos debido a la gran oscilatoriedad de los mismos.

En efecto, tal como se ve en el ejemplo, la integración de a_{10} del desarrollo de $f(x)$= ABS(sen x) con 5 cifras significativas, tardó 34 m 40 s

Tratando de evitar tan largas duraciones se ha intentado el método indicado por William M. Kahan ("padre" de la implantación de $\int_x^y$ en la HP-34C) en el artículo "Handheld Calculator Evaluates Integrals", aparecido en Hewlett-Packard Journal, August, 1980, pag. 31. Aplicado a este caso, tendríamos:

$$a_i = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(ix)\, dx = \frac{1}{\pi} \sum_{j=1}^{i} \int_{\frac{2\pi}{i}(j-1)}^{\frac{2\pi}{i}j} f(x) \cos(ix)\, dx$$

en la que se ha descompuesto el intervalo de integración en i subintervalos. Después, cambiando el sumatorio y la integral y haciendo la transformación ix = t llegaríamos a:

$$\frac{1}{\pi} \int_0^{2\pi} \cos(ix) \sum_{j=1}^{i} f(x)\, dx = \frac{1}{i\pi} \int_0^{2\pi} \cos t \sum_{j=1}^{i} f\left[\frac{t + 2\pi(j-1)}{i}\right] dt$$

Resuelto así el problema, la solución del a_{10} de antes tarda en llegar 39 m 6 s. ¿Por qué aumentó la duración?.

Repetidos los cálculos se pudo comprobar que la integración directa del coeficiente representaba la evaluación de 511 veces el integrando ABS(sen x) cos(10 x), mientras que el empleo del método sumatorio sólo lo evaluaba 63 veces. Sin embargo el sumatorio se extiende a 10 términos, lo que representa 630 veces la evaluación de ABS(sen x), además de que el cálculo del integrando es más largo puesto que ha de acumular la suma.

En definitiva, si bien el método de Mr. Kahan consigue la misma precisión con un considerable descenso del número de evaluaciones del integrando, en este caso no se traduce, en general, en una menor duración debido a la mayor complejidad de la evaluación.

Así pues, la solución adoptada es la de la integración directa según las fórmulas:

$$a_o = \frac{1}{2\pi} \int_0^{2\pi} f(x)\, dx \qquad a_i = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos(ix)\, dx$$

$$b_i = \frac{1}{\pi} \int_0^{2\pi} f(x) \operatorname{sen}(ix)\, dx \quad , \text{con } i = 1, 2, \ldots \text{ etc.}$$

3.- Observaciones.

Para discriminar entre el cálculo de a_i o b_i se ha recurrido al "flag" 0, que es puesto o borrado por las entradas A o B. Así, el programa se mantiene común a ambos cálculos salvo la evaluación de COS o SIN en la subrutina 1 del integrando que se decide según la situación del "flag" 0.

4.- Ejemplo.

Sea la función y = ABS(sen x), resultado de la rectificación ideal de onda completa.

Programada bajo 0 sería:

```
046   LBL 0
047   f SIN
048   h ABS
049   h RTN
```

Ajustando la notación a SCI 5 para obtener 6 cifras exactas resulta:

0; A	6,36620 E-1	(a_0)	(4 m 22 s)
2; A	-4,24413 E-1	(a_2)	(8 m 47 s)
4; A	-8,48826 E-2	(a_4)	(17 m 28 s)
10; A	-1,28610 E-2	(a_{10})	(34 m 40 s)

Los coeficientes a_i impares (i= 1, 3, ...) así como los b_i deberán ser nulos por ser la función de simetría par y completa.

El último caso calculado, a_{10}, se ha tomado como base de comparación entre los dos métodos descritos antes. Con el programa enlistado $\int_y^x$ calculó 511 veces el integrando. Sin embargo, todos los resultados fueron correctos en las 6 cifras mostradas. Por consiguiente, y antes de enjuiciar la integración directa, sería necesario probar con otro método (por ejemplo, el del programa anterior) y comparar la duración necesaria para obtener la misma exactitud, prescindiendo de que, por memoria, el número de coeficientes estaría limitado, y por método, realiza acumulaciones innecesarias para coeficientes nulos.

En resumen... es cuestión de no tener prisa.

HP-34C

TITULO: DESARROLLO DE FOURIER POR INTEGRACION DE COEFICIENTES.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA			
2	PROGRAMAR BAJO LBL 0 EL CÁLCULO DE $Y=F(x)$, SUPONIENDO A X EN EL REGISTRO X Y DEJANDO A Y EN EL MISMO REGISTRO. HAY DISPONIBLE DESDE 047 A 070 CON R_3 A $R_{.9}$, O BIEN DESDE 047 A 189 SIN NINGÚN REGISTRO. ADEMÁS SE DISPONE DE LBL 4 A LBL 9.			
3	AJUSTAR NOTACIÓN SEGÚN PRECISIÓN REQUERIDA		f sci	
4	CALCULAR COEFICIENTES a_i HACIENDO, PARA $i=0,1,$ 2, ... HASTA DONDE PROCEDA	i	A	a_i
5	CALCULAR COEFICIENTES b_i HACIENDO, PARA $i=1,2,$ 3, ... HASTA DONDE PROCEDA	i	B	b_i
6	ADEMÁS DE MOSTRARLOS EN PANTALLA, EN LOS PASOS 4 Y 5, a_i ó b_i QUEDAN ALMACENADOS EN R_0.			
7	PARA MODIFICAR LA EXACTITUD, IR A 3.			
8	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: DESARROLLO DE FOURIER POR INTEGRACION.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	×	1		6		1		6	
2	h SF 0	7	h F? 0	2		7		2		7	
3	GTO 2	8	GTO 2	3		8		3		8	
4	h LBL B	9	f SIN	4		9		4		9	
5	g $x=0$	040	×	5		110		5		180	
6	h RTN	1	h RTN	6		1		6		1	
7	h CF 0	2	h LBL 2	7		2		7		2	
8	h LBL 2	3	f COS	8		3		8		3	
9	g RAD	4	×	9		4		9		4	
010	STO 2	5	h RTN	080		5		150		5	
1	0	6	h LBL 0	1		6		1		6	
2	h π	7		2		7		2		7	
3	2	8		3		8		3		8	
4	×	9		4		9		4		9	
5	RCL 2	050		5		120		5		190	
6	g $x=0$	1		6		1		6		1	
7	GTO 2	2		7		2		7		2	
8	g R↓	3		8		3		8		3	
9	f $\int_y^x$ 1	4		9		4		9		4	
020	h LBL 3	5		090		5		160		5	
1	h π	6		1		6		1		6	
2	÷	7		2		7		2		7	
3	STO 0	8		3		8		3		8	
4	h RTN	9		4		9		4		9	
5	h LBL 2	060		5		130		5		200	
6	g R↓	1		6		1		6		1	
7	f $\int_y^x$ 0	2		7		2		7		2	
8	2	3		8		3		8		3	
9	÷	4		9		4		9		4	
030	GTO 3	5		100		5		170		5	
1	h LBL 1	6		1		6		1		6	
2	STO 1	7		2		7		2		7	
3	GSB 0	8		3		8		3		8	
4	RCL 1	9		4		9		4		9	
5	RCL 2	070		5		140		5		210	

Registros

0	a_i , b_i	1	x	2	i	3		4		5		6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6		.7		.8		.9		I	

Etiquetas

A : INTEGRA COEFICIENTES a_i

B : INTEGRA COEFICIENTES b_i

0 : CALCULA $f(x)$, FUNCION A DESARROLLAR.

1 : CALCULA $f(x) \cdot \sin ix$ ó $f(x) \cdot \cos ix$.- 2 y 3

Flags

0 : DECIDE SI CALCULA a_i ó b_i

Modo angular

RAD

Notacion

DESARROLLO DE CHEBYSHEV

1.- Objeto.

Dada la función contínua $y=f(x)$ en el intervalo $A \leq x \leq B$, el siguiente programa obtiene los coeficientes del desarrollo de Chebyshev dado por

$$y = d_o + d_1 \cos t + d_2 \cos 2t + \ldots + d_n \cos nt =$$
$$= d_o + d_1 T_1 + d_2 T_2 + \ldots + d_n T_n$$

en la que $T_i = \cos it$, para $u = \cos t$, son los polinomios ortogonales de Chebyshev, con $-1 \leq u \leq 1$. El programa calcula hasta d_{10}, y con $m > 20$ puntos de apoyo para $f(x)$, resulta un ajuste por mínimos cuadrados.

La expresión del desarrollo obtenida es en función de la variable unificada u. Con el programa 2.10, después de economizar si procede el desarrollo, se podrá obtener un polinomio en u. Por último, con el programa 2.11 se obtendrá el polinomio equivalente en la variable original x.

2.- Método.

Al tratar de ajustar o interpolar una función aperiódica cualquiera $y=f(x)$ mediante un desarrollo de Fourier, cabe suponer una falsa periodicidad en el intervalo considerado, equivalente a $0 \div 2\pi$, pero, salvo que casualmente se cumpla $y_o = y_m$, presentará discontinuidades en el origen, entorpeciendo la exactitud del ajuste y requiriendo mayor número de puntos de apoyo.

La solución del problema está en hacer un cambio de variable determinado. En primer lugar, si el intervalo en que se desea ajustar $f(x)$ es $A \div B$, debemos hacer $f(x) = g(u)$, con $-1 \leq u \leq 1$, y $A \equiv -1$ y $B \equiv 1$. Tras esto, si $u = \cos t$, el intervalo de $u(-1,1)$ pasa a ser $t(-\pi, \pi)$. En el nuevo sistema (t, y) la función $\overline{g}(t)$ se reproduce deformada dos veces, una en el intervalo $(-\pi, 0)$ y otra de $(0, \pi)$ siendo inevitable $\overline{g}(-\pi) = \overline{g}(\pi)$, eludiéndose así el problema de la discontinuidad.

El método a seguir es el mismo que en un desarrollo de Fourier, excepto el cambio $u = \cos t$. Además, por la simetría respecto al eje YY' resultante, los términos en sen son nulos.

Así, pues, el cálculo de los coeficientes d_i es similar al de los a_i de los programas precedentes, resultando:

$$d_0 = \frac{1}{\pi}\int_0^{\pi} y\, dt \qquad d_i = \frac{2}{\pi}\int_0^{\pi} y \cos it\, dt$$

en las que el intervalo de integración es $0 \div \pi$ por la simetría antes citada.

Igual que en el desarrollo de Fourier, la ortogonalidad de las funciones trigonométricas se extiende también a las sumas finitas, pudiéndose entonces poner:

$$d_0 = \frac{1}{m}\sum_{j=0}^{m-1} y_j \qquad d_i = \frac{2}{m}\sum_{j=0}^{m-1} y_j \cos it_j$$

en las que $y_j = \overline{g}(t_j)$, $u_j = \cos t_j$, y $t_j = \pi(2j/m - 1)$.

En resumen, el método seguido consiste en:

a) Dividir el intervalo $(-\pi, \pi)$ en m subintervalos a definir por el usuario. A mayor m, mejor aproximación. Se obtienen así los t_j según la fórmula anterior.

b) Con cada t_j calcular $u_j = \cos t_j$. Conocidos los límites A, B, obtener la x_j correspondiente a u_j mediante el cambio de variable necesario. Así se llega a $y_j = f(x_j)$.

c) Hacer las acumulaciones necesarias según las fórmulas sumatorias anteriores que nos dan los coeficientes buscados.

3.- Observaciones.

Por restricciones de almacenamiento el programa llega hasta d_{10}, si bien esto será suficiente en la mayoría de los casos. Normalmente, al empleo de este programa deberá seguir una economización, tal como se vió en los programas 2.9 y 2.10, mediante la cual se reduce el grado del polinomio ajustado al mínimo tolerable según el -- error fijado.

Así, pues, la utilidad de este cálculo es extraordinaria cuando se trata de obtener aproximaciones polinómicas a funciones cualesquiera, tal como se ve en el ejemplo.

4.- Ejemplo.

Se trata de aproximar, mediante un polinomio, la función exponencial en el intervalo -1, 1, y con un error máximo de 10^{-7}.

Programamos, bajo B, la función e^x y fijamos m= 24.

Haremos:

1; CHS; ENTER ↑ ; 1; ENTER ↑ ;

24; A .... 1,0000 (10 m 22 s)

Recuperamos ahora los coeficientes obteniados, con FIX 7:

RCL 0	1,2660659	(d_0)
RCL 1	1,1303182	(d_1)
RCL 2	0,2714953	(d_2)
RCL 3	0,0443368	(d_3)
RCL 4	0,0054742	(d_4)
RCL 5	0,0005429	(d_5)
RCL 6	0,0000450	(d_6)
RCL 7	0,0000032	(d_7)
RCL 8	0,0000002	(d_8)
RCL 9	1,414000 E-8	(d_9)
RCL.0	1,390000 E-9	(d_{10})

Con el error fijado, podremos despreciar los dos últimos coeficientes. Empleado el programa 2.10 obtenemos la siguiente expresión polinómica final:

$$e^x = 1 + 0,999999\ x + 0,500001\ x^2 + 0,166674\ x^3 + 0,041666\ x^4 + \\ + 0,008317\ x^5 + 0,00139\ x^6 + 0,00021\ x^7 + 0,00003\ x^8$$

válida para el intervalo $-1 \leqslant x \leqslant 1$.

HP-34C

TITULO: DESARROLLO DE CHEBYSHEV.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PROGRAMAR BAJO LBL B LA FUNCION A DESARROLLAR O APROXIMAR. SUPONER X EN EL REGISTRO X Y DEJAR Y=F(X) EN EL MISMO REGISTRO. HAY DISPONIBLES 7 PASOS DE PROGRAMA (DE 64 A 70) Y 4 REGISTROS (DE R.6 A R.9) Ó 35 PASOS (DE 64 A 98) Y NINGUN REGISTRO}.			
3	INGRESAR LIMITES DE LA VARIABLE:			
	- INFERIOR	A	ENTER ↑	
	- SUPERIOR	B	ENTER ↑	
4	INGRESAR NUMERO DE INTERVALOS ($M \geq 20$) E INICIAR (M HA DE SER PAR)	M	A	1,0000
5	OBTENER COEFICIENTES DE LA APROXIMACION DE GRADO 10 HACIENDO, PARA $k=0,\ldots,10$		RCL [k]	d_k
6	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA: DESARROLLO DE CHEBYSHEV.-

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	2	1		6		1		6	
2	g RAD	7	×	2		7		2		7	
3	f REG	8	ENTER↑	3		8		3		8	
4	STO . 3	9	ENTER↑	4		9		4		9	
5	STO f I	040	ENTER↑	5		110		5		180	
6	g R↓	1	1	6		1		6		1	
7	STO . 2	2	.	7		2		7		2	
8	g R↓	3	0	8		3		8		3	
9	STO . 1	4	1	9		4		9		4	
010	h LBL 1	5	f x⇄I	080		5		150		5	
1	1	6	STO . 5	1		6		1		6	
2	RCL f I	7	h LBL 2	2		7		2		7	
3	2	8	CLX	3		8		3		8	
4	×	9	RCL . 4	4		9		4		9	
5	RCL . 3	050	RCL f I	5		120		5		190	
6	÷	1	h INT	6		1		6		1	
7	−	2	×	7		2		7		2	
8	h π	3	f COS	8		3		8		3	
9	×	4	×	9		4		9		4	
020	STO . 4	5	STO + (i)	090		5		160		5	
1	f COS	6	g ISG	1		6		1		6	
2	RCL . 2	7	GTO 2	2		7		2		7	
3	RCL . 1	8	RCL . 5	3		8		3		8	
4	−	9	STO f I	4		9		4		9	
5	×	060	g DSE	5		130		5		200	
6	RCL . 2	1	GTO 1	6		1		6		1	
7	+	2	h RTN	7		2		7		2	
8	RCL . 1	3	h LBL B	8		3		8		3	
9	+	4		9		4		9		4	
030	2	5		100		5		170		5	
1	÷	6		1		6		1		6	
2	GSB B	7		2		7		2		7	
3	RCL . 3	8		3		8		3		8	
4	÷	9		4		9		4		9	
5	STO + 0	070		5		140		5		210	

Registros

0	d_0	1	d_1	2	d_2	3	d_3	4	d_4	5	d_5	6	d_6
7	d_7	8	d_8	9	d_9	.0	d_{10}	.1	A	.2	B	.3	M
.4	t	.5	Utiliz.	.6		.7		.8		.9		I	i

Etiquetas

A : CALCULO DE COEFICIENTES DEL DESARROLLO.

B : FUNCION A DESARROLLAR (A PROGRAMAR)

1 y 2.

Flags

Modo angular

RAD

Notacion

DESARROLLO DE CHEBYSHEV POR INTEGRACION DE COEFICIENTES

1.- Objeto.

Dada la función contínua $y = f(x)$, definida en el intervalo $A \leqslant x \leqslant B$, este programa obtiene, uno a uno, los coeficientes del desarrollo de Chebyshev dado por

$$y = d_o + d_1T_1 + d_2T_2 + \ldots + d_nT_n$$

con n hasta donde proceda. La expresión así obtenida está en la variable unificada $-1 \leqslant u \leqslant 1$, siendo necesarios los programa 2.10 y 2.11 para obtener la aproximación polinómica a $f(x)$.

2.- Método.

Estamos en la misma situación que en el programa 3.4. Aquí, el desarrollo del método indicado por Mr. Kahan llevaría a la expresión:

$$d_i = \frac{2}{\pi i} \int_0^{\pi} \cos t \sum_{j=1}^{i} (-1)^{j-1} f(u)\, dt$$

$$\text{con } u = \cos \frac{t + \pi(j-1)}{i}$$

Como antes, se ha efectuado la prueba comparativa entre este método y la integración directa de los coeficientes, resultando una duración mayor en un 80% para aquel.

En consecuencia se ha elegido la integración directa de los coeficientes según

$$d_o = \frac{1}{\pi} \int_0^{\pi} y\, dt \qquad\qquad d_i = \frac{2}{\pi} \int_0^{\pi} y \cos it\, dt$$

haciendo el programa todas las transformaciones de variables necesarias.

3.- Observaciones.

Por el propio objetivo del programa, la obtención de los d_i se efectúa uno a uno pero secuencialmente. El usuario se detendrá cuando ~~cuando~~ comience a obtener coeficientes inferiores al ~~e~~ error prefijado. La precisión obtenida dependerá de la notación ajustada, que es con la que trabajará $\int_y^x$.

4.- Ejemplo.

Se desea aproximar polinómicamente e^x en el intervalo --- $-1 \leq x \leq 1$. Programada la función bajo LBL B, haremos:

1; CHS; ENTER↑ ; 1; A	0,00000	
X⇄Y	1,26607	(d_o)
R/S	1,00000	
X⇄Y	1,13032	(d_1)
R/S	2,00000	
X⇄Y	0,27150	(d_2)
ETC.		

El tiempo de cálculo es de alrededor de 5 m para cada coeficiente con el FIX 5 elegido. Sin embargo, leyendo los coeficientes con FIX 7 (fijando después nuevamente FIX 5 para el siguiente cálculo), obtenemos:

$d_o = 1,2660659$

$d_1 = 1,1303182$

$d_2 = 0,2714954$

Etc.

Resultan más próximos a los verdaderos que lo que cabría pensar ya que el margen de error que resulta de $\int \frac{x}{y}$ es mejor que las cifras ajustadas. Por la propia naturaleza del empleo de este programa -- convendrá, en general, trabajar con FIX y no con SCI, lo que acorta el tiempo de ejecución.

HP-34C

TITULO: DESARROLLO DE CHEBYSHEV POR INTEGRACION DE COEFICIENTES.

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PROGRAMAR BAJO LBL B EL CÁLCULO DE $Y=F(X)$, SUPONIENDO A X EN EL REGISTRO X Y DEJANDO A Y EN EL MISMO REGISTRO. HAY DISPONIBLES DESDE 048 A 070 CON R_5 A $R_{.9}$ Ó BIEN DESDE 048 A 175 SIN NINGÚN REGISTRO. ADEMÁS DE LBL 4 A LBL 9.			
3	AJUSTAR NOTACIÓN SEGÚN PRECISIÓN REQUERIDA.			
4	INGRESAR LÍMITE INFERIOR	A	ENTER↑	
	LIMITE SUPERIOR	B		
	E INICIAR		A	0
5	OBTENER COEFICIENTE DE ORDEN 0		$x \rightleftarrows y$	d_0
6	PARA SUCESIVOS COEFICIENTES HACER, PARA $k=1, 2, \ldots$ HASTA DONDE SEA NECESARIO		R/S	k
			$x \rightleftarrows y$	d_k
7	PARA UN NUEVO CASO, IR A 2.			

NOTA: EN EL PASO 6, ADEMÁS DE SER PRESENTADO EN PANTALLA d_k, QUEDA ALMACENADO EN R_0 HASTA QUE d_{k+1} SEA CALCULADO.

SI SE TIENE NECESIDAD DE APAGAR LA CALCULADORA, SE PODRÁ REANUDAR EL CÁLCULO, TRAS SU PUESTA EN "ON", CON GSB 3.

HP-34C PROGRAMA : DESARROLLO DE CHEBYSHEV POR INTEGRACIÓN

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	RCL 4	1		6		1		6	
2	g RAD	7	x	2		7		2		7	
3	STO 1	8	f cos	3		8		3		8	
4	STO 2	9	x	4		9		4		9	
5	g R↓	040	h RTN	5		110		5		180	
6	STO − 1	1	h LBL 0	6		1		6		1	
7	STO −2	2	f cos	7		2		7		2	
8	2	3	RCL 1	8		3		8		3	
9	STO ÷ 1	4	x	9		4		9		4	
010	STO ÷2	5	RCL 2	080		5		150		5	
1	CLX	6	+	1		6		1		6	
2	STO 4	7	h LBL B	2		7		2		7	
3	h π	8		3		8		3		8	
4	f $\int_Y^x$ 0	9		4		9		4		9	
5	GTO 2	050		5		120		5		190	
6	h LBL 3	1		6		1		6		1	
7	g RAD	2		7		2		7		2	
8	1	3		8		3		8		3	
9	STO + 4	4		9		4		9		4	
020	0	5		090		5		160		5	
1	h π	6		1		6		1		6	
2	f $\int_Y^x$ 1	7		2		7		2		7	
3	2	8		3		8		3		8	
4	x	9		4		9		4		9	
5	h LBL 2	060		5		130		5		200	
6	h π	1		6		1		6		1	
7	÷	2		7		2		7		2	
8	STO 0	3		8		3		8		3	
9	RCL 4	4		9		4		9		4	
030	R/S	5		100		5		170		5	
1	GTO 3	6		1		6		1		6	
2	h LBL 1	7		2		7		2		7	
3	STO 3	8		3		8		3		8	
4	GSB 0	9		4		9		4		9	
5	RCL 3	070		5		140		5		210	

Registros

0	d_i	1	k_1	2	k_0	3	t	4	i	5		6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6		.7		.8		.9		I	

Etiquetas

A : INICIA CALCULO.

B : CALCULA Y = F(x). A PROGRAMAR POR EL USUARIO.

0, 1, 2 Y 3

Flags

Modo angular

RAD

Notacion

SOLUCION DE ECUACIONES DIFERENCIALES DE PRIMER ORDEN

1.- Objeto.

Dada la ecuación diferencial de 1º orden $y' = f(x,y)$ y los valores iniciales $y(x_0) = y_0$, el siguiente programa obtiene aproximaciones numéricas a la solución $y(x) = y$.

2.- Método.

Para generar las aproximaciones a $y_n = y(x_0 + nh)$, siendo h un tamaño de paso fijado por el usuario y con n= 1, 2, ..., se emplea la fórmula abierta de Runge-Kutta de 4º orden:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

siendo:

$$k_1 = h\ f(x_n,\ y_n)$$
$$k_2 = h\ f(x_n + h/2,\ y_n + k_1/2)$$
$$k_3 = h\ f(x_n + h/2,\ y_n + k_2/2)$$
$$k_4 = h\ f(x_n + h,\ y_n + k_3).$$

Aún cuando en la integración de un paso se ha de calcular $f(x,\ y)$ cuatro veces, la precisión del resultado es notable, siendo proporcional el error a h^5.

3.- Observaciones.

A pesar de la aparente complejidad del método, se ha podido reducir el programa con el empleo del salto indicado. A través -- del registro I el programa salta a la etiqueta correcta que prepara a las variables x e y para el siguiente cálculo. Si bien esto supone que las variables x e y han de estar almacenadas, sólo representa un pequeño trabajo accesorio a la hora de ingresar los valores iniciales.

4.- Ejemplo.

Sea la ecuación diferencial

$$y' = 1/x^2 - y/x - y^2$$

con $x_0 = 1$ e $y_0 = -1$.

El programa se prueba con h igual a 0,1, 0,05 y 0,025, lo que dará idea del efecto del paso.

Los resultados obtenidos se tabulan seguidamente:

Con h = 0,1 :

x	y	y`
1,1	-0,909089993	0,826447114
1,2	-0,833331802	0,694445720
1,3	-0,769228788	0,591717500
1,4	-0,714283377	0,510205751
1,5	-0,666664029	0,444446203
⋮		
2,0	-0,499996197	0,250001902

Con h = 0,05

1,1	-0,909090849	0,826446335
1,2	-0,833333233	0,694444528
1,3	-0,769230640	0,591716076
1,4	-0,714285562	0,510204190
1,5	-0,666666495	0,444444559
⋮		
2,0	-0,499999752	0,250000124

Con h = 0,025

1,1	-0,909090905	0,826446285
1,2	-0,833333327	0,694444450
1,3	-0,769230761	0,591715983
1,4	-0,714285705	0,510204089
1,5	-0,666666656	0,444444451
⋮		
2,0	-0,499999985	0,250000008

Evidentemente ha sido necesario efectuar dos pasos de integración por cada valor tabulado, y cuatro pasos, para los tamaños 0,05 y 0,025 respectivamente.

HP-34C

TITULO: SOLUCION DE ECUACIONES DIFERENCIALES DE PRIMER GRADO.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PROGRAMAR, BAJO LBL B, LA ECUACION $Y'=f(x,y)$ SUPONIENDO A X EN R_0 E Y EN R_1. DEJAR Y' EN EL REGISTRO X. HAY DISPONIBLES DE 050 A 070 CON R_5 A $R_{.9}$ Ó DE 050 A 154 SIN NINGÚN REGISTRO.			
3	AJUSTAR MODO ANGULAR, SI PROCEDE.			
4	ALMACENAR VALORES INICIALES	x_0	STO 0	
		Y_0	STO 1	
5	ALMACENAR TAMAÑO DEL PASO	h	STO 2	
6	INICIAR E INTEGRAR UN PASO		A	x
			$x \rightleftarrows y$	y
7	PARA OBTENER EL VALOR CORRESPONDIENTE DE Y'		B	y'
8	PARA SUCESIVOS VALORES $x_i = x_0 + ih$ Y CORRESPONDIENTES DE Y_i E Y_i', REPETIR 6 Y 7 PARA $i = 1,2,\ldots$ HASTA DONDE SEA NECESARIO.			
9	PARA OTRO TAMAÑO DE PASO, IR A 4.			
10	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA : SOL. DE ECUACIONES DIF. DE 1ER GRADO.-

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	h LBL 3	1		6		1		6	
2	RCL 1	7	STO + 4	2		7		2		7	
3	STO 3	8	STO + 4	3		8		3		8	
4	5	9	GTO 4	4		9		4		9	
5	STO f I	040	h LBL 1	5		110		5		180	
6	CLx	1	RCL 4	6		1		6		1	
7	STO 4	2	6	7		2		7		2	
8	h LBL 4	3	÷	8		3		8		3	
9	GSB B	4	RCL 3	9		4		9		4	
010	RCL 2	5	+	080		5		150		5	
1	×	6	STO 1	1		6		1		6	
2	STO + 4	7	RCL 0	2		7		2		7	
3	2	8	h RTN	3		8		3		8	
4	÷	9	h LBL B	4		9		4		9	
5	ENTER↑	050		5		120		5		190	
6	ENTER↑	1		6		1		6		1	
7	RCL 3	2		7		2		7		2	
8	+	3		8		3		8		3	
9	STO 1	4		9		4		9		4	
020	g R↓	5		090		5		160		5	
1	g DSE	6		1		6		1		6	
2	GTO f I	7		2		7		2		7	
3	h LBL 4	8		3		8		3		8	
4	RCL 2	9		4		9		4		9	
5	2	060		5		130		5		200	
6	÷	1		6		1		6		1	
7	STO + 0	2		7		2		7		2	
8	GTO 4	3		8		3		8		3	
9	h LBL 2	4		9		4		9		4	
030	STO + 1	5		100		5		170		5	
1	RCL 2	6		1		6		1		6	
2	2	7		2		7		2		7	
3	÷	8		3		8		3		8	
4	STO + 0	9		4		9		4		9	
5	g R↓	070		5		140		5		210	

Registros

0	x	1	Y	2	h	3	Utiliz.	4	Utiliz.	5		6	
7		8		9		.0		.1		.2		.3	
.4		.5		.6		.7		.8		.9		I	i

Etiquetas

A : OBTIENE LA INTEGRACION DE UN PASO.

B : CALCULA $Y' = f(x,Y)$. A PROGRAMAR POR EL USUARIO.

1, 2, 3 y 4.

Flags

Modo angular

Notacion

SOLUCION DE SISTEMAS DE DOS ECUACIONES DIFERENCIALES DE 1º ORDEN Y DE ECUACIONES DIFERENCIALES DE 2º ORDEN.-

1.- Objeto.

Dado el sistema de dos ecuaciones diferenciales de 1º orden:

$$y' = f(x, y, z)$$
$$z' = g(x, y, z)$$

y las condiciones iniciales $y(x_o)=y_o$; $z(x_o)=z_o$, el siguiente programa calcula aproximaciones numéricas a las soluciones:

$$y = y(x) \quad ; \quad z = z(x)$$

para intervalos de la variable x definidos por el usuario.

Así mismo, se puede emplear para obtener aproximaciones a la solución de ecuaciones diferenciales de 2º orden:

$$y'' = f(x, y, y')$$

mediante la equivalencia:

$$y' = z$$
$$z' = y'' = f(x, y, z).$$

2.- Método.

Se emplea la fórmula abierta de Runge-Kutta de 4º orden, que aplicada a un sistema sería:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \qquad z_{n+1} = z_n + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4)$$

$$k_1 = h\ f(x_n, y_n, z_n) \qquad m_1 = h\ g(x_n, y_n, z_n)$$

$$k_2 = h\ f(x_n + h/2, y_n + k_1/2, z_n + m_1/2); \quad m_2 = h\ g(x_n + h/2, y_n + k_1/2, z_n + m_1/2)$$

$$k_3 = h\ f(x_n + h/2, y_n + k_2/2, z_n + m_2/2); \quad m_3 = h\ g(x_n + h/2, y_n + k_2/2, z_n + m_2/2)$$

$$k_4 = h\ f(x_n + h, y_n + k_3, z_n + m_3) \qquad m_4 = h\ g(x_n + h, y_n + k_3, z_n + m_3).$$

El tamaño del paso, h, lo define el usuario, teniendo en cuenta que el error es proporcional a h^5.

3.- Observaciones.

El programa es idéntico al anterior en cuanto a su estructura, salvo la duplicidad de cálculos por ser dos ecuaciones. Nuevamente, el empleo del salto indicado, reduce considerablemente la extensión del programa.

4.- Ejemplos.

a) Sea el sistema $y' = y + z$

$z' = z - y$

con $y(0) = 0{,}1$; $z(0) = 0{,}2$.

Tras programar y\` y z\` bajo A y B, empleamos el programa con dos tamaños de paso:

Con h = 0,1

x	y	z	y\`	z\`
0,1	0,13203167	0,20889667	0,34092833	0,07686500
0,2	0,16823689	0,21514573	0,38338261	0,04690884
0,3	0,20873944	0,21802286	0,42676230	0,00928342
0,4	0,25359551	0,21671792	0,47031343	-0,03687759
⋮				
1,0	0,60434264	0,06500178	0,66934442	-0,53934085

Con h = 0,05

x	y	z	y\`	z\`
0,1	0,13203157	0,20889664	0,34092821	0,07686507
0,2	0,16823667	0,21514568	0,38338235	0,04690901
0,3	0,20873907	0,21802282	0,42676190	0,00928375
0,4	0,25359497	0,21671792	0,47031289	-0,03687706
0,5	0,30277677	0,21033389	0,51311066	-0,09244288
⋮				
1,0	0,60434059	0,06500316	0,66934375	-0,53933743

b) Sea la ecuación de 2º orden $y'' = -y$, con las condiciones iniciales $y(0)=0$ e $y'(0)=1$.

Puesto que la equivalencia a considerar es $y'=z$ y ------ $z'=y''= -y$, las dos funciones se programarán simplemente:

```
h LBL A          h LBL B
  RCL 2            RCL 1
h RTN              CHS
                 h RTN
```

e ingresaremos: 0; STO 0; STO 1; 1; STO 2. Empleando h = 0,1 ------ (0,1; STO 3) resulta:

x	y	z=y\`
0,1	0,099833	0,995004
0,2	0,198669	0,980067
0,3	0,295520	0,955337

Ya que la solución es y = sen x, se observará que las 6 cifras son exactas. El tiempo de integración de un paso es de unos 20 s.

HP-34C

TITULO: SOLUCION DE UN SISTEMA DE 2 ECUACIONES DIFERENCIALES DE 1ER GRADO.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PROGRAMAR LAS DOS ECUACIONES: BAJO LBL A, $y' = f(x,y,z)$; BAJO LBL B, $z' = g(x,y,z)$, SUPONIENDO A X EN R_0, Y EN R_1 Y Z EN R_2. HAY DISPONIBLES DE R_9 A $R_{.8}$ Ó 70 PASOS DE PROGRAMA PARA AMBAS.			
3	AJUSTAR MODO ANGULAR, SI PROCEDE.			
4	ALMACENAR VALORES INICIALES	x_0	STO 0	
		y_0	STO 1	
		z_0	STO 2	
5	ALMACENAR TAMAÑO DE PASO	h	STO 3	
6	INICIAR LA INTEGRACION		GSB 0	x_1
			g R↓	y_1
			g R↓	z_1
7	PARA OBTENER LOS VALORES DE LAS DERIVADAS HACER		A	y_1'
			B	z_1'
8	PARA LA INTEGRACION DE SUCESIVOS PASOS TAL QUE $x_i = x_0 + ih$, PARA $i = 2, \ldots$ HASTA DONDE SEA NECESARIO, HACER		R/S	x_i
			g R↓	y_i
			g R↓	z_i
			A	y_i'
			B	z_i'
9	PARA MODIFICAR TAMAÑO DE PASO, IR A 4.			
10	PARA UN NUEVO CASO, IR A 2.			

HP-34C

PROGRAMA : SISTEMAS DE 2 ECUAC. DIF. DE 1ER GRADO.-

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL 0	6	g DSE	1	RCL 4	6		1		6	
2	CLX	7	GTO f I	2	STO 1	7		2		7	
3	STO 6	8	h LBL 4	3	RCL 0	8		3		8	
4	STO 7	9	RCL 3	4	h RTN	9		4		9	
5	RCL 1	040	2	5	h LBL A	110		5		180	
6	STO 4	1	÷	6	h RTN	1		6		1	
7	RCL 2	2	STO + 0	7	h LBL B	2		7		2	
8	STO 5	3	GTO 4	8		3		8		3	
9	5	4	h LBL 2	9		4		9		4	
010	STO f I	5	STO + 2	080		5		150		5	
1	h LBL 4	6	RCL 8	1		6		1		6	
2	GSB A	7	STO + 1	2		7		2		7	
3	RCL 3	8	CLX	3		8		3		8	
4	×	9	RCL 3	4		9		4		9	
5	STO + 6	050	2	5		120		5		190	
6	2	1	÷	6		1		6		1	
7	÷	2	STO + 0	7		2		7		2	
8	STO 8	3	g R↓	8		3		8		3	
9	GSB B	4	h LBL 3	9		4		9		4	
020	RCL 3	5	STO + 7	090		5		160		5	
1	×	6	STO + 7	1		6		1		6	
2	STO + 7	7	RCL 8	2		7		2		7	
3	2	8	STO + 6	3		8		3		8	
4	÷	9	STO + 6	4		9		4		9	
5	ENTER↑	060	GTO 4	5		130		5		200	
6	ENTER↑	1	h LBL 1	6		1		6		1	
7	RCL 5	2	6	7		2		7		2	
8	+	3	STO ÷ 6	8		3		8		3	
9	STO 2	4	STO ÷ 7	9		4		9		4	
030	CLX	5	RCL 6	100		5		170		5	
1	RCL 8	6	STO + 4	1		6		1		6	
2	RCL 4	7	RCL 7	2		7		2		7	
3	+	8	STO + 5	3		8		3		8	
4	STO 1	9	RCL 5	4		9		4		9	
5	g R↓	070	STO 2	5		140		5		210	

Registros

0	x	1	y	2	z	3	h	4	Utiliz.	5	Utiliz.	6	Utiliz.
7	Utiliz.	8	Utiliz.	9		.0		.1		.2		.3	
.4		.5		.6		.7		.8		.9		I	¿

Etiquetas

A : CALCULA $Y' = f(X, Y, Z)$. A PROGRAMAR POR EL USUARIO.

B : CALCULA $Z' = g(X, Y, Z)$. ID. ID. ID.

0 : INTEGRA UN PASO.

1, 2, 3 Y 4.

Flags

Modo angular

Notacion

SOLUCION DE ECUACIONES DIFERENCIALES DE 3º ORDEN.-

1.- Objeto.

Dada la ecuación diferencial de 3º orden

$$y''' = f(x,\ y,\ y',\ y'')$$

y las condiciones iniciales $y(x_o)= y_o$; $y'(x_o)= y'_o$; $y''(x_o)= y''_o$, -- el siguiente programa obtiene aproximaciones numéricas a la solución $y = y(x)$ para intervalos de la variable x definidos por el usuario.

2.- Método.

En esta ocasión se emplea la fórmula abierta de Runge-Kutta de 2º orden. En realidad, la ecuación de 3º orden se puede transformar en un sistema de tres ecuaciones de 1º orden, y, aplicando la formulación de Runge-Kutta, se obtienen las fórmulas de integración:

$$y_{n+1}= y_n+ \frac{h}{2}(2y'_n + hy''_n)$$

$$y'_{n+1}= y'_n+ \frac{h}{2}(2y''_n + hk_1)$$

$$y''_{n+1}= y''_n+ \frac{h}{2}(k_1 + k_2)$$

$$k_1= f(x_n,\ y_n,\ y'_n,\ y''_n)$$

$$k_2= f(x_n+ h,\ y_n+ hy'_n,\ y'_n+hy''_n,\ y''_n + hk_1)$$

Al tratarse de la fórmula de 2º orden, el error es proporcional a h^3, y por lo tanto de menor exactitud que las empleadas en los anteriores programas. O lo que es igual, será necesario emplear tamaños más pequeños para h para conseguir una exactitud equivalente.

Debido a esto, se prevé la integración de n pasos ininterrumpidamente, siendo definido n por el usuario.

La razón de haber escogido la fórmula de 2º orden es por la larga extensión de un programa para la de 4º orden.

3.- Observaciones.

No es necesario aquí recurrir al salto indicado. El programa calcula sucesivamente la función definida. El usuario define el tamaño del paso y el número de pasos que desea calcular sin interrupción, es decir, sin visualizar las aproximaciones. Esto permite trabajar con pasos pequeños sin necesidad de arrancar el programa tras aproximaciones que no interesan.

4.- <u>Ejemplo</u>.

Sea la ecuación diferencial de 3º orden

$$y''' = -2(y + y')$$

con las condiciones iniciales $y(0)=0$; $y'(0)=0$; $y''(0)=2$.

Se programa, pues:

```
h LBL B
  RCL 1
  RCL 2
  -
  2
  x
  CHS
h RTN
```

Y se almacenan: 0; STO 0; STO 1; STO 2; 2; STO 3.

Eligiendo h=0,01 y n=10 (0,01; STO 4; 10; STO 5) se obtiene:

x	y	y'	y''	y'''
0,1	0,009983655	0,199324578	1,979373310	-0,418616466
0,2	0,039726197	0,394438077	1,915249086	-0,868328548
0,3	0,088584294	0,580851596	1,805018917	-1,338871781
0,4	0,155454431	0,753868995	1,647169083	-1,818646853
0,5	0,238757873	0,908702722	1,441404383	-2,294921190

El tiempo de cálculo de cada valor es de 1 m aproximadamente.

HP-34C

TITULO: SOLUCION DE ECUACIONES DIFERENCIALES DE 3ER GRADO.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	PROGRAMAR BAJO LBL B EL CALCULO DE $Y''' = f(X, Y, Y', Y'')$, SUPONIENDO A X EN R_0, Y EN R_1, Y' EN R_2 E Y'' EN R_3. DEJAR Y''' EN EL REGISTRO X. HAY DISPONIBLES DE 053 A 070 CON R_7 A $R_{.9}$ Ó DESDE 053 A 143 SIN NINGUN REGISTRO.			
3	AJUSTAR MODO ANGULAR, SI PROCEDE.			
4	ALMACENAR VALORES INICIALES	X_0	STO 0	
		Y_0	STO 1	
		Y_0'	STO 2	
		Y_0''	STO 3	
5	ALMACENAR TAMAÑO DE PASO Y NUMERO DE INTERVALOS	h	STO 4	
		N	STO 5	
6	PARA VALORES SUCESIVOS CORRESPONDIENTES A $x_i = x_0 + i h N$, HACER, PARA $i = 1, 2, \ldots$ HASTA DONDE SEA NECESARIO		A	x_i
			g R↓	Y_i
			g R↓	Y_i'
			g R↓	Y_i''
			B	Y_i'''
7	PARA OTRO TAMAÑO DE PASO Y/O NUMERO DE INTERVALOS, IR A 4.			
8	PARA UN NUEVO CASO, IR A 2.			

HP-34C PROGRAMA : SOLUCION DE ECUACIONES DIFERENC. DE 3ER GRADO.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL A	6	h LSTX	1		6		1		6	
2	RCL 5	7	RCL 6	2		7		2		7	
3	STO f I	8	×	3		8		3		8	
4	h LBL 0	9	STO + 2	4		9		4		9	
5	GSB B	040	RCL 4	5		110		5		180	
6	STO 6	1	×	6		1		6		1	
7	RCL 4	2	STO − 1	7		2		7		2	
8	STO + 0	3	g R↓	8		3		8		3	
9	RCL 2	4	STO + 3	9		4		9		4	
010	×	5	g DSE	080		5		150		5	
1	STO + 1	6	GTO 0	1		6		1		6	
2	g R↓	7	RCL 3	2		7		2		7	
3	RCL 4	8	RCL 2	3		8		3		8	
4	RCL 3	9	RCL 1	4		9		4		9	
5	×	050	RCL 0	5		120		5		190	
6	STO + 2	1	h RTN	6		1		6		1	
7	g R↓	2	h LBL B	7		2		7		2	
8	RCL 4	3		8		3		8		3	
9	×	4		9		4		9		4	
020	STO + 3	5		090		5		160		5	
1	GSB B	6		1		6		1		6	
2	RCL 6	7		2		7		2		7	
3	−	8		3		8		3		8	
4	RCL 4	9		4		9		4		9	
5	2	060		5		130		5		200	
6	÷	1		6		1		6		1	
7	×	2		7		2		7		2	
8	h LSTX	3		8		3		8		3	
9	RCL 4	4		9		4		9		4	
030	×	5		100		5		170		5	
1	RCL 3	6		1		6		1		6	
2	x⇄y	7		2		7		2		7	
3	×	8		3		8		3		8	
4	STO + 1	9		4		9		4		9	
5	g R↓	070		5		140		5		210	

Registros

0	x	1	y	2	y'	3	y''	4	h	5	N	6	Utiliz.
7		8		9		.0		.1		.2		.3	
.4		.5		.6		.7		.8		.9		I	k

Etiquetas

A : INTEGRA UN PASO.

B : CALCULA $y''' = f(x, y, y', y'')$. A PROGRAMAR POR EL USUARIO.-

0 .

Flags

Modo angular

Notacion

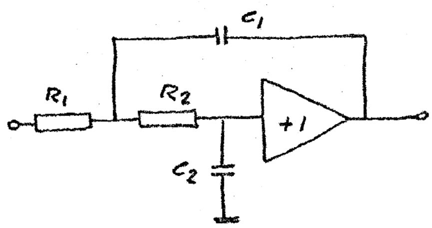
FILTROS ACTIVOS SALLEN-KEY DE 2º ORDEN

1.- Objeto.

Obtener los valores necesarios de resistencias y condensadores para realizar filtros activos pasa-bajo y pasa-alto, de dos --polos, con estructura de Sallen-Key.

2.- Método.

La estructura de Sallen-Key emplea un amplificador con ganancia +1, y en su versión pasa-bajo de dos polos adopta la configuración del esquema. La función de transferencia es:



$$T(p) = \frac{1}{p^2R_1R_2C_1C_2 + pC_2(R_1 + R_2) + 1}$$

Con ella se pueden realizar las funciones clásicas de Butterworth, Chebyshev, Hermite, etc. La función de transferencia generalizada será:

$$T(s) = \frac{1}{s^2a_2 + sa_1 + 1}$$

en la que a_2 y a_1 son los coeficientes definidos por la función elegida. Introduciendo el factor de escala para reducir la expresión a la frecuencia de corte, $s = p/w_o$, resulta:

$$T(p) = \frac{1}{p^2\ a_2/w_o^2 + p\ a_1/w_o + 1}$$

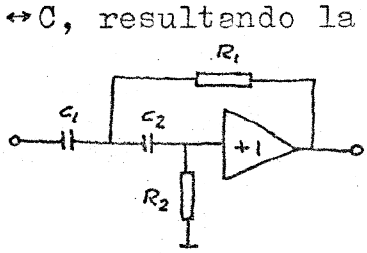
Igualando ambas expresiones se puede formar el sistema:

$$R_1R_2C_1C_2 = a_2/w_o^2$$

$$C_2(R_1 + R_2) = a_1/w_o$$

con dos ecuaciones y cuatro incógnitas, por lo que siempre se podrán elegir dos de ellas según valores normalizados comercializados.

Un filtro pasa-alto se obtendrá con la sustitución dual $R \leftrightarrow C$, resultando la configuración del esquema.



Su función de transferencia es:

$$T(p) = \frac{1}{p^2R_1R_2C_1C_2 + pR_1(C_1 + C_2) + 1}$$

Si en esta expresión se realiza la sustitución:

$R_1=1/C'_1$; $R_2=1/C'_2$; $C_1=1/R'_1$; $C_2=1/R'_2$; $p=1/p'$; resulta:

$$T(p')= \frac{1}{p'^2R'_1R'_2C'_1C'_2 + p'C'_2(R'_1 + R'_2) + 1}$$

que coincide con la del pasa-bajo. La solución es, pues, la misma, salvo la sustitución realizada, en la que $p'=1/p$ se cumple con --- $w'_o=1/w_o$.

El método de resolución se inicia adoptando valores para R_1 y R_2. Si se cumple que $R_1=R_2=R$, C_1 y C_2 siempre tendrán soluciones reales positivas:

$$C_1= \frac{a_2}{w_oa_1}\,\frac{R_1 + R_2}{R_1R_2} \quad (1); \qquad C_2= \frac{a_1}{w_o(R_1+ R_2)} \quad (2)$$

Si, por el contrario, fijamos los valores de C_1 y C_2, podemos obtener: $R_1R_2= a_2/w_o^2C_1C_2$; $R_1+ R_2= a_1/w_oC_2$; con lo que

$$(R_1,\ R_2) = \frac{a_1}{2w_oC_2}\left(1 \pm \sqrt{1 - 4a_2C_2/a_1^2C_1}\right) \quad (3)$$

Para que R_1, R_2 tengan soluciones reales positivas se deberá cumplir

$$\sqrt{1 - 4a_2C_2/a_1^2C_1} < 1 \ ; \quad 1 - 4a_2C_2/a_1^2C_1 \geqslant 0$$

que nos lleva a: $a_2C_2/a_1^2C_1 > 0$; $C_2 \leqslant a_1^2C_1/4a_2$ (4)

La primera condición siempre se cumple por ser todos los factores positivos. Luego sólo se ha de considerar la segunda.

Resumiendo todo lo anterior, se sigue el siguiente procedimiento:

1) Fijar valores, generalmente normalizados, para R_1 y R_2. Normalmente se intentará $R_1=R_2$.

2) Calcular C_1 según expresión (1). Si el valor resultante es facilmente obtenible con elementos normalizados, adoptarlo. Si no, introducir el valor normal más próximo al calculado.

3) Con el valor adoptado para C_1, calcular el límite de C_2, según expresión (4). Si es facilmente obtenible, adoptar el C_2 calculado. Si no, adoptar un valor normal inferior al calculado.

4) Definidos así C_1 y C_2, calcular R_1 y R_2 por (3).

Según lo expuesto, si se parten de valores $R_1=R_2=R$ normalizados, los C_1 y C_2 resultantes se tendrán que obtener, en general, por combinación de dos o más condensadores. Si se fijan en 2) y 3) valores normalizados para C_1 y C_2, los valores resultantes para

R_1 y R_2 no lo serán, y se deberá buscar una combinación que proporcione el valor resultante.

3.- <u>Observaciones.</u>

Con objeto de lograr que el mismo programa calcule filtros pasa-alto y pasa-bajo, se ha recurrido a la subrutina indicada. De esta forma se emplea, en los casos en que el cálculo lo requiere, el valor directo o el inverso. Resulta ser una solución más económica que el empleo de "flags" o cualquier otra, ya que la llamada a la subrutina sólo emplea un paso de programa, mientras que la prueba -- de un "flag" requiere dos.

4.- <u>Ejemplos.</u>

a) Se desea construir un filtro Sallen-Key de 3º orden, pasa-bajo, para una frecuencia de corte de 1500 kHz. La respuesta deseada es de Butterworth.

La función de transmisión, según lo anterior, es:

$$1/T(s) = p^3 + 2p^2 + 2p + 1 = (p^2 + p + 1)(p + 1)$$

Para la sección de 2º orden deberá ser, pues, $a_1=1$ y $a_2=2$.

Haremos:

1; STO.1; STO.2
1.500; GSB 0 .... 9,425E3 (w_o)

Probando con valores $R_1=R_2=R=10$ k , resulta:

10 E3; ENTER↑ ; B .... 21,22 E-9 (C_1^*)
(C_1) 22 E-9; R/S .... 5,500 E-9 (C_2^*)
(C_2) 4,7 E-9; R/S .... 15,59 E3 (R_1)
X⇄Y .... 6,983 E3 (R_2)

Para conseguir la combinación de resistencias adecuadas, hacemos:

15,59 E3; ENTER↑ ; 18 E3; GSB 2 .... 116,4 E3
6,983 E3; ENTER↑ ; 8,2 E3; GSB 2 .... 47,04 E3

La solución final podría ser:

R_1= 18 kΩ//120 kΩ R_2= 8,2 kΩ//47 kΩ
C_1= 22 nF C_2= 4,7 nF

La célula pasiva de 1º orden se calculará:

(R) 10 E3; GSB 1 .... 10,61 E-9 (C)

Luego se puede prever: R= 10 kΩ C= 10 nF

b) Se desea construir un filtro Sallen-Key, de 2º orden, pasa-alto, con respuesta de Chebyshev de 0,5 dB de rizado, para una frecuencia de corte de 10 kHz.

La respuesta elegida tiene como función de transmisión

$$1/T(p) = 0{,}6595p^2 + 0{,}9402p + 1$$

luego, $a_1 = 0{,}9402$ y $a_2 = 0{,}6595$.

Hacemos:

0,9402; STO.1; 0,6595; STO.2

(f_0) 10.000; GSB 0 .... 62,83 E3 (w_0)

Probamos con $C_1 = C_2 = C =$ 1 nF, por lo que:

(C) 1 E-9; ENTER↑ ; A .... 11,34 E3 (R_1^*)

(R_1) 10 E3; R/S .... 29,84 E3 (R_2^*)

(R_2) 33 E3; R/S .... 783,6 E-12 (C_1)

X⇄Y .... 1,485 E-9 (C_2)

Buscando la combinación de C_1 haremos:

783,6 E-12; ENTER↑ ; 820 E-12; GSB 2 .... 17,65 E-9

Tendremos finalmente:

$R_1 =$ 10 kΩ $R_2 =$ 33 kΩ

$C_1 =$ 820 pF//18 nF $C_2 =$ 1,5 nF

<u>NOTA</u>: VER ID. PAGINA 5.2.3.

HP-34C

TITULO: FILTROS ACTIVOS SALLEN-KEY DE 1ER Y 2º ORDEN.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR COEFICIENTES SEGUN TIPO DE RESPUESTA DESEADA (BUTTERWORTH, CHEBYSHEV, ETC)	a_1	STO . 1	
		a_2	STO . 2	
3	INGRESAR FRECUENCIA DE CORTE	f_0 (Hz)	GSB 0	ω_0
4	PARA CELULA PASIVA DE 1ER ORDEN, PASA-ALTO O PASA-BAJO, HACER	R (Ω) ó C (F)	GSB 1	C (F) ó R (Ω)
5	PARA 2º ORDEN PASA-BAJO, HACER:			
5.1	INGRESAR ESTIMACIONES DE RESISTENCIAS (NORMALMENTE $R_1 = R_2$)	R_1 (Ω)	ENTER ↑	
		R_2 (Ω)	B	C_1^* (F)
5.2	INGRESAR EL VALOR NORMALIZADO MÁS PRÓXIMO A C_1^* (SI NO SE DESEA, HACER $C_1 = C_1^*$)	C_1 (F)	R/S	C_2^* (F)
5.3	INGRESAR EL VALOR NORMALIZADO MÁS PRÓXIMO A C_2^* QUE CUMPLA $C_2 \leqslant C_2^*$ (SI NO, HACER $C_2 = C_2^*$)	C_2 (F)	R/S	R_1 (Ω)
			x⇄y	R_2 (Ω)
6	PARA 2º ORDEN PASA-ALTO, HACER:			
6.1	INGRESAR ESTIMACIONES DE CAPACIDADES (NORMALMENTE $C_1 = C_2$)	C_1 (F)	ENTER ↑	
		C_2 (F)	A	R_1^* (Ω)
6.2	INGRESAR EL VALOR NORMALIZADO MÁS PRÓXIMO A R_1^* (SI NO SE DESEA AJUSTAR, HACER $R_1 = R_1^*$)	R_1 (Ω)	R/S	R_2^* (Ω)
6.3	INGRESAR EL VALOR NORMALIZADO MÁS PRÓXIMO A R_2^* QUE CUMPLA $R_2 \geqslant R_2^*$ (SI NO, HACER $R_2 = R_2^*$)	R_2 (Ω)	R/S	C_1 (F)
			x⇄y	C_2 (F)
7	PARA OTROS INTENTOS CON DISTINTOS ESTIMADOS, IR A 4, 5 ó 6.			
8	PARA CONSEGUIR COMBINACIONES EN PARALELO DE RESISTENCIAS (O EN SERIE DE CONDENSADORES):			
8.1	INGRESAR VALOR A CONSEGUIR	R ó C	ENTER ↑	
8.2	INGRESAR UN VALOR NORMALIZADO TAL QUE R' > R	R' ó C'	GSB 2	R'' ó C''
8.3	SE CUMPLIRÁ R = R'//R'' ó C = C'//C''. REPETIR CON DISTINTOS R' ó C' HASTA QUE R'' ó C'' SEA SATISFACTORIO.			
9	PARA OTRAS FRECUENCIAS, IR A 3.			
10	PARA OTRAS RESPUESTAS, IR A 2.			

HP-34C PROGRAMA : FILTROS ACTIVOS SALLEN-KEY DE 1º Y 2º ORDEN.-

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL 0	6	RCL .2	1	−	6		1		6	
2	f ENG 3	7	×	2	f √x	7		2		7	
3	h π	8	RCL .1	3	STO + 1	8		3		8	
4	×	9	÷	4	STO − 2	9		4		9	
5	2	040	RCL 0	5	RCL .1	110		5		180	
6	×	1	GSB f I	6	2	1		6		1	
7	STO 0	2	÷	7	÷	2		7		2	
8	h RTN	3	GSB f I	8	RCL 0	3		8		3	
9	h LBL 1	4	R/S	9	GSB f I	4		9		4	
010	h 1/x	5	GSB f I	080	÷	5		150		5	
1	RCL 0	6	STO 3	1	RCL 4	6		1		6	
2	÷	7	RCL .1	2	÷	7		2		7	
3	h RTN	8	g x²	3	STO × 1	8		3		8	
4	h LBL 2	9	×	4	STO × 2	9		4		9	
5	h 1/x	050	RCL .2	5	RCL 2	120		5		190	
6	x⇄y	1	÷	6	GSB f I	1		6		1	
7	h 1/x	2	4	7	RCL 1	2		7		2	
8	x⇄y	3	÷	8	GTO f I	3		8		3	
9	−	4	GSB f I	9	h LBL 5	4		9		4	
020	h 1/x	5	R/S	090	h 1/x	5		160		5	
1	h RTN	6	GSB f I	1	h LBL 4	6		1		6	
2	h LBL A	7	STO 4	2		7		2		7	
3	5	8	RCL .2	3		8		3		8	
4	GTO 3	9	×	4		9		4		9	
5	h LBL B	060	4	5		130		5		200	
6	4	1	×	6		1		6		1	
7	h LBL 3	2	RCL .1	7		2		7		2	
8	STO f I	3	g x²	8		3		8		3	
9	g R↓	4	÷	9		4		9		4	
030	GSB f I	5	RCL 3	100		5		170		5	
1	h 1/x	6	÷	1		6		1		6	
2	x⇄y	7	1	2		7		2		7	
3	GSB f I	8	STO 1	3		8		3		8	
4	h 1/x	9	STO 2	4		9		4		9	
5	+	070	x⇄y	5		140		5		210	

Registros

0	w_0	1	R_1 , $1/C_1$	2	R_2 , $1/C_2$	3	C_1 , $1/R_1$	4	C_2 , $1/R_2$	5		6	
7		8		9		.0		.1	a_1	.2	a_2	.3	
.4		.5		.6		.7		.8		.9		I	i

Etiquetas

A : CALCULA FILTRO PASA-ALTO ; B : ID. ID. PASA-BAJO.

0 : CALCULA $w_0 = 2\pi f_0$; 1 : CALCULA SECCION 1ER ORDEN.

2 : CALCULA R'' CONOCIDOS R Y R' (R=R'//R'').

3, 4 Y 5.

Flags

Modo angular

Notacion

ENG 3

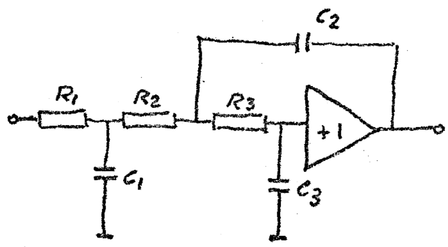
FILTROS ACTIVOS SALLEN-KEY DE 3º ORDEN

1.- Objeto.

Obtener los valores necesarios de resistencias y condensadores para realizar filtros activos pasa-bajo y pasa-alto, de tres polos, con estructura de Sallen-Key.

2.- Método.

Se puede realizar un filtro de 3º orden con una sección de 2 polos y una célula pasiva de 1º orden. O bien se puede sintetizar en un solo elemento activo, adoptando la configuración que presenta la figura para el caso pasa-bajo.



La función de transmisión, para este último caso, sería:

$$1/T(p)=p^3R_1R_2R_3C_1C_2C_3 + p^2(C_2C_3(R_1R_3 + R_2R_3) + C_1C_3(R_1R_3 + R_1R_2)) + $$
$$+ p(C_3(R_1 + R_2 + R_3) + R_1C_1) + 1$$

El filtro pasa-alto se consigue con la sustitución dual de $R \leftrightarrow C$. Si en su función de transferencia se efectúa la sustitución:

$$R_1' = 1/C_1;\ R_2' = 1/C_2;\ R_3' = 1/C_3;\ C_1' = 1/R_1;\ C_2' = 1/R_2;\ C_3' = 1/R_3;\ p' = 1/p$$

resulta la misma función que para el pasa-bajo, por lo que sólo se tratará este caso, previendo el programa la utilización de ambos.

La función de transferencia generalizada de 3º orden es:

$$T(s)=1/(s^3a_3 + s^2a_2 + sa_1 + 1)$$

y, con la transformación $p=s/w_o$, resulta:

$$T(p)=1/(p^3a_3/w_o^3 + p^2a_2/w_o^2 + pa_1/w_o + 1)$$

que, igualada a la del filtro forma un sistema de tres ecuaciones con seis incógnitas, habiendo libertad para elegir tres de ellas.

El sistema sería:

$$R_1R_2R_3C_1C_2C_3 = a_3/w_o^3$$

$$C_3(R_3C_2(R_1 + R_2) + R_1C_1(R_2 + R_3)) = a_2/w_o^2$$

$$C_3(R_1 + R_2 + R_3) + R_1C_1 = a_1/w_o$$

Partiendo de $R_1=R_2=R_3=R$ se llega siempre a valores reales positivos para C_1, C_2 y C_3, por lo que habitualmente se empieza -- asignando valores a las resistencias.

Haciendolo así se llega a la ecuación cúbica:

$$R_1^2(R_2+R_3)C_1^3/(R_1+R_2+R_3) - a_1R_1(R_2+R_3)C_1^2/w_o(R_1+R_2+R_3) +$$
$$+ a_2C_1/w_o^2 - a_3(1/R_1+1/R_2)/w_o^3 = 0 \qquad (1)$$

Resolviéndola y obteniendo C_1, podemos calcular:

$$C_2= a_3/w_o^3R_1R_2R_3C_1C_3 \quad (2) \;; \qquad C_3=(a_1/w_o - C_1R_1)/(R_1+R_2+R_3) \quad (3)$$

Si definimos las siguientes variables auxiliares:

$$C_o=a_1/w_oR_1;\; A_1=(R_1+R_2+R_3)/R_1;\; A_2= R_1(R_2+R_3)$$
$$A_3= w_o^3R_1R_2R_3C_1C_3$$

las ecuaciones (1), (2) y (3) pasan a ser

$$A_2C_1^3/A_1 - C_oA_2C_1^2/A_1+ a_2C_1/w_o^2 - a_3(1/R_1+1/R_2)/w_o^3 = 0 \qquad (4)$$
$$C_2= a_3/A_3 \quad (5) \;; \qquad C_3=(C_o - C_1)/A_1 \qquad (6)$$

La ecuación (4) puede ponerse en la forma:

$$C_1= C_o+ (a_3(1/R_1+1/R_2)/w_o - C_oa_2)/(A_2(C_1w_o)^2/A_1+ a_2)$$

que puede resolverse por medio de una iteración de punto fijo, con el valor de arranque $C_1=C_o$.

Después de esto, se podrá calcular C_2 y C_3 aplicando (5) y (6).

Otra posibilidad sería asignar valores a C_1, C_2 y C_3 y -- calcular las resistemcias, pero es más laborioso puesto que hay que partir de las ecuaciones implícitas, no habiendo capacidad para ello. Tampoco es posible recurrir al doble cálculo del programa anterior, igualmente por falta de capacidad.

3.- Observaciones.

Igual que antes, se ha usado la subrutina indicada para calcular el filtro pasa-bajo, tal como se desarrolla en el método, como el filtro pasa-alto. El cálcúlo iterativo de C_1 requiere, además, mayor duración de ejecución.

4.- Ejemplos.

a) Se desea construir un filtro pasa-alto de 3º orden, con respuesta de Butterworth, para una frecuencia de corte de 1500 Hz.

La función de transmisión es:

$1/T(p) = 1 + 2p + 2p^2 + p^3$; es decir: $a_1=2$; $a_2= 2$; $a_3= 1$.

Haremos, pues:

2; STO.1; STO.2; 1; STO.3

1.500; GSB 0 .... 9,425 E3

(C_1) 10 E-9; ENTER↑ ; ENTER↑ ; A .... 7,619 E3 (R_1)

g R↓ .... 2,991 E3 (R_2)

g R↓ .... 52,41 E3 (R_3)

Utilizando GSB 1 llegamos a:

R_1= 10 kΩ//33 kΩ $C_1=C_2=C_3$= 10 nF (asignado)

R_2= 3,3 kΩ//33 kΩ

R_3= 56 kΩ//820 kΩ

b) Se desea ahora un filtro pasa-bajo de 3º orden, con -- respuesta de Chebyshev 0,1 dB, para 15 kHz.

Según tablas, las constantes son:

a_1= 1,6046 ; a_2= 1,1832 ; a_3= 0,6101

Almacenadas en STO.1, STO.2 y STO.3, respectivamente, seguiremos haciendo:

15 E3; GSB 0 .... 94,25 E3

(R_1) 10 E3; ENTER↑ ; ENTER↑ ; B .... 1,394 E-9 (C_1)

g R↓ .... 5,087 E-9 (C_2)

g R↓ .... 102,8 E-12 (C_3)

Recurriendo a GSB 1 para los condensadores, resulta:

C_1= 1,5 nF + 20 nF ; C_2= 5,6 nF + 56 nF ; C_3= 100 pF.

Por asignación, $R_1=R_2=R_3$= 10 k .

Nota: El método empleado está inspirado en el artículo "Empleo del Calculador en el Diseño de Filtros Activos", de A. Jugs, de RCA Engineer, publicado en MUNDO ELECTRONICO, nº 38, 1975.

HP-34C

TITULO: FILTROS ACTIVOS SALLEN-KEY DE 3ER ORDEN.-

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	ALMACENAR COEFICIENTES SEGUN TIPO DE RES-	a_1	STO . 1	
	PUESTA DESEADA (BUTTERWORTH, CHEBYSHEV, ETC.)	a_2	STO . 2	
		a_3	STO . 3	
3	INGRESAR FRECUENCIA DE CORTE	f_0 (Hz)	GSB 0	ω_0
4	PARA FILTRO PASA-BAJO, SEGUIR. EN OTRO CASO,			
	IR A 6.			
5	SI ES FILTRO PASA-BAJO, INGRESAR VALORES ESTI-	R_1 (Ω)	ENTER ↑	
	MADOS DE RESISTENCIAS (NORMALMENTE $R_1=R_2=R_3$)	R_2 (Ω)	ENTER ↑	
		R_3 (Ω)	B	C_1 (F)
			g R↓	C_2 (F)
			g R↓	C_3 (F)
6	SI ES FILTRO PASA-ALTO, INGRESAR VALORES ESTIMA-	C_1 (F)	ENTER ↑	
	DOS DE CAPACIDADES (NORMALMENTE $C_1=C_2=C_3$)	C_2 (F)	ENTER ↑	
		C_3 (F)	A	R_1 (Ω)
			g R↓	R_2 (Ω)
			g R↓	R_3 (Ω)
7	PARA OTROS INTENTOS CON DISTINTOS ESTIMADOS, IR			
	A 5 ó 6.			
8	PARA CONSEGUIR COMBINACIONES EN PARALELO DE RE-			
	SISTENCIAS (O EN SERIE DE CONDENSADORES), HACER			
8.1	INGRESAR VALOR A CONSEGUIR	R ó C	ENTER ↑	
8.2	INGRESAR VALOR NORMALIZADO TAL QUE R'>R	R' ó C'	GSB 1	R" ó C"
8.3	SE CUMPLIRÁ R = R'//R" ó C = C'//C". REPETIR CON			
	DISTINTOS R' ó C' HASTA QUE R" ó C" SEA SATISFAC-			
	TORIO.			
9	PARA OTRAS FRECUENCIAS, IR A 3.			
10	PARA OTRAS RESPUESTAS, IR A 2.			

HP-34C PROGRAMA: FILTROS ACTIVOS SALLEN-KEY DE 3ER ORDEN.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL 0	6	g R↓	1	GSB f I	6	GSB f I	1		6	
2	f ENG 3	7	GSB f I	2	×	7	RCL 4	2		7	
3	h π	8	STO 1	3	g x^2	8	GTO f I	3		8	
4	×	9	STO + 7	4	RCL 8	9	h LBL 5	4		9	
5	2	040	STO ÷ 7	5	×	110	h 1/x	5		180	
6	×	1	STO × 8	6	RCL 7	1	h LBL 4	6		1	
7	STO 0	2	STO × 9	7	÷	2		7		2	
8	h RTN	3	RCL 0	8	RCL .2	3		8		3	
9	h LBL 1	4	GSB f I	9	+	4		9		4	
010	h 1/x	5	STO × 9	080	÷	5		150		5	
1	x⇄y	6	STO × 9	1	RCL 6	6		1		6	
2	h 1/x	7	STO × 9	2	+	7		2		7	
3	x⇄y	8	×	3	STO − 4	8		3		8	
4	−	9	RCL .1	4	RCL 4	9		4		9	
5	h 1/x	050	x⇄y	5	x⇄y	120		5		190	
6	h RTN	1	÷	6	STO 4	1		6		1	
7	h LBL A	2	STO 4	7	÷	2		7		2	
8	5	3	STO 6	8	h ABS	3		8		3	
9	GTO 2	4	h LBL 3	9	EEX	4		9		4	
020	h LBL B	5	RCL 1	090	4	5		160		5	
1	4	6	h 1/x	1	CHS	6		1		6	
2	h LBL 2	7	RCL 2	2	f x≤y	7		2		7	
3	STO f I	8	h 1/x	3	GTO 3	8		3		8	
4	g R↓	9	+	4	RCL 4	9		4		9	
5	GSB f I	060	RCL .3	5	STO × 9	130		5		200	
6	STO 3	1	×	6	STO − 6	1		6		1	
7	STO 7	2	RCL 0	7	RCL 7	2		7		2	
8	STO 8	3	GSB f I	8	STO ÷ 6	3		8		3	
9	STO 9	4	÷	9	RCL 6	4		9		4	
030	g R↓	5	RCL 6	100	STO × 9	5		170		5	
1	GSB f I	6	RCL .2	1	GSB f I	6		1		6	
2	STO 2	7	×	2	RCL .3	7		2		7	
3	STO + 7	8	−	3	RCL 9	8		3		8	
4	STO + 8	9	RCL 4	4	÷	9		4		9	
5	STO × 9	070	RCL 0	5	STO 5	140		5		210	

Registros

0	ω_0	1	R_1, $1/C_1$	2	R_2, $1/C_2$	3	R_3, $1/C_3$	4	C_1, $1/R_1$	5	C_2, $1/R_2$	6	C_3, $1/R_3$
7	AUX 1	8	AUX 2	9	AUX 3	.0		.1	a_1	.2	a_2	.3	a_3
.4		.5		.6		.7		.8		.9		I	

Etiquetas

A: CALCULA FILTRO PASA-ALTO.
B: CALCULA FILTRO PASA-BAJO.
0: CALCULA $\omega_0 = 2\pi f_0$; 1: CALCULA R'', CONOCIDOS R Y R' TAL QUE $R = R' // R''$; 2,3,4 Y 5.

Flags

Modo angular

Notacion

ENG 3

EQUILIBRADO DINAMICO EN DOS PLANOS

1.- Objeto.

El siguiente programa calcula los pesos correctores necesarios, a disponer en dos planos del rotor de una máquina rotativa, dadas las vibraciones a corregir y conocidas las características del rotor. Si éstas no se conociesen, permite obtenerlas mediante dos pruebas con pesos conocidos.

2.- Método.

Sea una máquina rotativa en la que se miden las siguientes vibraciones, dadas en coordenadas polares:

VO1: vibración en el plano 1.

VO2: vibración en el plano 2.

En el supuesto más general en que hayan de obtenerse las características mediante pruebas, sea PO1 un peso arbitrario colocado en el plano 1 y medido su módulo y posición angular, resultará así mismo en coordenadas polares. Tras su colocación, las nuevas vibraciones medidas son:

VP1: vibración en el plano 1.

VC2: vibración en el plano 2.

En una segunda prueba, se quita PO1 y se coloca PO2 en el plano 2 (de hecho, puede ser el mismo peso pero, en general, en distinta posición, por lo que PO1≠PO2). Esto produce:

VC1: vibración en el plano 1.

VP2: vibración en el plano 2.

En el supuesto de que el rotor cumple la ley de Hook, y recurriendo al principio de superposición, se pueden establecer las siguientes relaciones vectoriales entre efecto y cáusa, es decir, entre vibraciones y pesos que las producen:

RP1=(VP1 - VO1)/PO1 : efecto neto de PO1 en el plano 1.

RP2=(VP2 - VO2)/PO2 : efecto neto de PO2 en el plano 2.

RC1=(VC1 - VO1)/PO2 : efecto neto de PO2 en el plano 1.

RC2=(VC2 - VO2)/PO1 : efecto neto de PO1 en el plano 2.

Si ahora deseamos anular las vibraciones existentes en los planos 1 y 2, que en general llamaremos V1 y V2, (inicialmente serán las mismas VO1 y VO2 ya utilizadas) podremos conocer los pesos P1 y P2 a colocar en los planos respectivos resolviendo el siguiente sistema vectorial:

$$P1\ RP1 + P2\ RC1 = -V1$$
$$P1\ RC2 + P2\ RP2 = -V2$$

en el que la inversión de signo en el miembro de la derecha se explica por el hecho de tener que producir, con P1 y P2, vibraciones opuestas a las existentes.

Por comodidad de manipulación de signos, haciendo cada --- R= -R, se llegaría al mismo sistema de antes pero con todos los términos positivos. Desde luego los nuevos R se obtendrían con numeradores opuestos a los expresados antes.

Resuelto el sistema, conoceremos P1 y P2, es decir, la magnitud y la situación angular de los pesos a colocar en los planos 1 y 2 para anular ambas vibraciones.

En la práctica, por inexactitudes de medidas y por no ajustarse a la realidad la hipótesis, no se conseguirá anular totalmente la vibración. Supongamos VR1 y VR2 las vibraciones medidas residuales después de la corrección. Se deberá recalcular nuevos pesos para su eliminación y para ello se pueden seguir dos procedimientos:

1) Resolver nuevamente el sistema con los mismos coeficientes pero cambiando el miembro de la derecha, sustituyéndolo por las nuevas vibraciones, haciendo, pues, V1=VR1 y V2=VR2. Esto equivale a dar como definitiva la corrección ya efectuada. Entonces, los nuevos pesos correctores que resulten se añadirán a los ya existentes en el rotor. (Este método es el empleado por la casa IRD).

2) La composición de la vibración inicial con la producida por los pesos no ha sido nula, quedando un resíduo, VR1 y VR2. Si éste lo agregamos a la vibración inicial y resolvemos el sistema para la vibración total, deberíamos corregir simultáneamente ambas. Es decir, si resolvemos nuevamente el sistema para

$$V1= V1 + VR1 \qquad y \qquad V2= V2 + VR2$$

los nuevos pesos resultantes deberán sustituir a los anteriores en magnitud y situación. Ello supone el desmontar los pesos tras cada reajuste para colocar los nuevos.

Ambos procedimientos son algebráicamente equivalentes. Se podrá seguir el que dé lugar a la operación más cómoda.

En cuanto a las magnitudes, es indistinto las unidades empleadas para vibración y pesos, así como si aquella es de desplazamiento o de velocidad. Lo que hay que prever es que sean las mismas dentro de un mismo cálculo.

Con relación a la situación angular tampoco debe haber problema. Normalmente se emplarán grados sexagesimales. La escala de referencia estará fija en el estator, no importando el sentido en que se midan los ángulos con tal que sea el mismo dentro del mismo cálculo. Como resultado del cálculo, el peso corrector deberá situarse en una posición angular determinada. Para ello, un sistema podría ser situando la marca de referencia ~~fija~~ del rotor sobre el 0 de la escala angular fija en el estator. El peso deberá colocarse en el punto del rotor (en el plano que corresponda) que coincida con al ángulo resultante del cálculo leído en la escala fija del estator.

3.- Observaciones.

El método de solución del sistema es la regla de Cramer, -- similar al empleado en el programa 1.4. El programa dispone de las subrutinas necesarias para realizar la aritmética compleja necesaria para su utilización.

En el caso de disponer ya de los coeficientes R, por haber efectuado alguna prueba anterior sobre la misma máquina, se podrán realizar directamente los pasos 3.4, 3.8, 3.12 y 3.16 de las instrucciones.

4.- Ejemplo.

En una determinada máquina se leen las siguientes vibraciones: V1= 110 nm$_{\lfloor 86º}$ V2= 74 nm$_{\lfloor 255º}$

Siendo necesario efectuar pruebas para la determinación de las reacciones del rotor, se coloca en el plano 1 el peso PO1= 150 g$_{\lfloor 0º}$, leyendose entonces

VP1= 153 nm$_{\lfloor 45º}$ VC2= 85 nm$_{\lfloor 200º}$

Puesto el mismo peso y en la misma situación en el plano 2 resultan VC1= 90 nm$_{\lfloor 90º}$ VP2= 120 nm$_{\lfloor 180º}$

Según lo expuesto, deberá ser VO1=V1 y VO2=V2. Hacemos:

(VP1) 45; ENTER ↑ ; 153; ENTER ↑

(VO1) 86; ENTER ↑ ; 110; GSB 7 100,5260

(PO1)	0; ENTER↑ ; 150; GSB 1	0,6702	
	STO 0; X⇄Y; STO 1	179,1195	(RP1)
(VP2)	180; ENTER↑ ; 120; ENTER↑		
(VO2)	255; ENTER↑ ; 74; GSB 7	123,6098	
(PO2)	0; ENTER↑ ; 150; GSB 1	0,8241	
	STO 6; X⇄Y; STO 7	-35,3282	(RP2)
(VC1)	90; ENTER↑ ; ENTER↑		
(VO1)	86; ENTER↑ ; 110; GSB 7	21,1715	
(PO2)	0; ENTER↑ ; 150; GSB 1	0,1411	
	STO 2; X⇄Y; STO 3	68,7504	(RC1)
(VC2)	200; ENTER↑ ; 85; ENTER↑		
(VO2)	255; ENTER↑ ; 74; GSB 7	74,0635	
(PO1)	0; ENTER↑ ; 150; GSB 1	0,4938	
	STO 4; X⇄Y; STO 5	-34,9299	(RC2)
	GSB 0	0,5798	(mod. Δ)
(V1)	110; STO 8; 86; STO 9		
(V2)	74; STO.0; 255; STO.1		
	A	76,6475	
	X⇄Y	56,6959	(P1)
	B	17,9756	
	X⇄Y	164,5756	(P2)

Los pesos a colocar serán:

P1 = 77 g $\underline{|57^{\circ}}$

P2 = 18 g $\underline{|165^{\circ}}$

HP-34C

TITULO: EQUILIBRADO DINAMICO EN DOS PLANOS.-

Instrucciones de uso

NUM.	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
1	CARGAR PROGRAMA.			
2	AJUSTAR MODO ANGULAR			
			g DEG	
3	FORMAR Y ALMACENAR COEFICIENTES :			
3.1	INGRESAR VIBRAC. PROPIA, PLANO 1	VP1 (arg)	ENTER ↑	
		VP1 (mod)	ENTER ↑	
3.2	INGRESAR VIBRAC. INICIAL, PLANO 1	VO1 (arg)	ENTER ↑	
		VO1 (mod)	GSB 7	
3.3	INGRESAR PESO DE PRUEBA, PLANO 1	PO1 (arg)	ENTER ↑	
		PO1 (mod)	GSB 1	RP1 (mod)
3.4	ALMACENAR COEFICIENTE		STO 0	
			x ⇄ y	RP1 (arg)
			STO 1	
3.5	INGRESAR VIBRAC. PROPIA, PLANO 2	VP2 (arg)	ENTER ↑	
		VP2 (mod)	ENTER ↑	
3.6	INGRESAR VIBRAC. INICIAL, PLANO 2	VO2 (arg)	ENTER ↑	
		VO2 (mod)	GSB 7	
3.7	INGRESAR PESO DE PRUEBA, PLANO 2	PO2 (arg)	ENTER ↑	
		PO2 (mod)	GSB 1	RP2 (mod)
3.8	ALMACENAR COEFICIENTE		STO 6	
			x ⇄ y	RP2 (arg)
			STO 7	
3.9	INGRESAR VIBRAC. CRUZADA, PLANO 1	VC1 (arg)	ENTER ↑	
		VC1 (mod)	ENTER ↑	
3.10	INGRESAR VIBRAC. INICIAL, PLANO 1	VO1 (arg)	ENTER ↑	
		VO1 (mod)	GSB 7	
3.11	INGRESAR PESO DE PRUEBA, PLANO 2	PO2 (arg)	ENTER ↑	
		PO2 (mod)	GSB 1	RC1 (mod)
3.12	ALMACENAR COEFICIENTE		STO 2	
			x ⇄ y	RC1 (arg)
			STO 3	
3.13	INGRESAR VIBRAC. CRUZADA, PLANO 2	VC2 (arg)	ENTER ↑	
		VC2 (mod)	ENTER ↑	
3.14	INGRESAR VIBRAC. INICIAL, PLANO 2	VO2 (arg)	ENTER ↑	
		VO2 (mod)	GSB 7	
3.15	INGRESAR PESO DE PRUEBA, PLANO 1	PO1 (arg)	ENTER ↑	
		PO1 (mod)	GSB 1	RC2 (mod)
3.16	ALMACENAR COEFICIENTE		STO 4	
			x ⇄ y	RC2 (arg)
			STO 5	
4	CALCULAR DETERMINANTE		GSB 0	Δ (arg)

HP-34 C

TITULO: EQUILIBRADO DINAMICO EN DOS PLANOS.- (CONTINUACIÓN)

Instrucciones de uso

NUM	OBSERVACIONES	ENTRADA DATOS	FUNCION	SALIDA RESULTADOS
5	ALMACENAR VIBRAC. A CORREGIR:			
5.1	DEL PLANO 1 (INICIALMENTE V1 = V01)	V1 (mod)	STO 8	
		V1 (arg)	STO 9	
5.2	DEL PLANO 2 (INICIALMENTE V2 = V02)	V2 (mod)	STO . 0	
		V2 (arg)	STO . 1	
6	CALCULAR PESO CORRECTOR EN EL PLANO 1		A	P1 (mod)
			x⇄y	P1 (arg)
7	CALCULAR PESO CORRECTOR EN EL PLANO 2		B	P2 (mod)
			x⇄y	P2 (arg)
8	EFECTUADA LA CORRECCION, TERMINAR EL EQUILIBRADO			
	SI LAS VIBRACIONES RESIDUALES, VR1 Y VR2, ESTÁN			
	DENTRO DE LA TOLERANCIA. EN OTRO CASO, SEGUIR.			
9	SI SE SIGUE EL 1ER PROCEDIMIENTO DE RECORRECCIÓN			
	(VER TEXTO), HACER V1 = VR1 Y V2 = VR2 Y VOLVER			
	A 5.			
10	SI SE SIGUE EL 2º PROCEDIMIENTO, OBTENER LAS			
	NUEVAS VIBRACIONES A CORREGIR Y ALMACENARLAS			
	HACIENDO:		RCL 9	V1 (arg)
			RCL 8	V1 (mod)
10.1	INGRESAR VIBRAC. RESIDUAL, PLANO 1	VR1 (arg)	ENTER ↑	
		VR1 (mod)	GSB 6	V1* (mod)
10.2	ALMACENAR NUEVAS VIBR. A CORREGIR, PLANO 1		STO 8	
			x⇄y	V1* (arg)
			STO 9	
			RCL . 1	V2 (arg)
			RCL . 0	V2 (mod)
10.3	INGRESAR VIBRAC. RESIDUAL, PLANO 2	VR2 (arg)	ENTER ↑	
		VR2 (mod)	GSB 6	V2* (mod)
10.4	ALMACENAR NUEVA VIBRAC. A CORREGIR, PLANO 2.		STO . 0	
			x⇄y	V2* (arg)
			STO . 1	
11	VOLVER A 6.			

HP-34C PROGRAMA: EQUILIBRADO DINAMICO EN DOS PLANOS.

Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION	Nº	FUNCION
001	h LBL 0	6	GSB 4	1	0	6		1		6	
2	RCL 1	7	RCL 9	2	−	7		2		7	
3	RCL 0	8	RCL 8	3	x⇄y	8		3		8	
4	RCL 7	9	RCL 5	4	GTO 5	9		4		9	
5	RCL 6	040	RCL 4	5	h LBL 7	110		5		180	
6	GSB 2	1	h LBL 3	6	GSB 4	1		6		1	
7	GSB 4	2	GSB 2	7	g R↓	2		7		2	
8	RCL 3	3	GSB 5	8	h LBL 5	3		8		3	
9	RCL 2	4	RCL .2	9	f →R	4		9		4	
010	RCL 5	5	÷	080	RCL .4	5		150		5	
1	RCL 4	6	x⇄y	1	x⇄y	6		1		6	
2	GSB 2	7	RCL .3	2	−	7		2		7	
3	GSB 5	8	−	3	STO .4	8		3		8	
4	STO .2	9	x⇄y	4	x⇄y	9		4		9	
5	g R↓	050	h RTN	5	RCL .5	120		5		190	
6	STO .3	1	h LBL 2	6	x⇄y	1		6		1	
7	h RTN	2	x⇄y	7	−	2		7		2	
8	h LBL A	3	g R↓	8	STO .5	3		8		3	
9	RCL 9	4	×	9	x⇄y	4		9		4	
020	RCL 8	5	g R↓	090	g →P	5		160		5	
1	RCL 7	6	+	1	h RTN	6		1		6	
2	RCL 6	7	f R↑	2	h LBL 1	7		2		7	
3	GSB 2	8	h RTN	3	h 1/x	8		3		8	
4	GSB 4	9	h LBL 4	4	x⇄y	9		4		9	
5	RCL 3	060	f →R	5	CHS	130		5		200	
6	RCL 2	1	STO .4	6	x⇄y	1		6		1	
7	RCL .1	2	g R↓	7	GTO 2	2		7		2	
8	RCL .0	3	STO .5	8		3		8		3	
9	GTO 3	4	h RTN	9		4		9		4	
030	h LBL B	5	h LBL 6	100		5		170		5	
1	RCL 1	6	GSB 4	1		6		1		6	
2	RCL 0	7	g R↓	2		7		2		7	
3	RCL .1	8	x⇄y	3		8		3		8	
4	RCL .0	9	1	4		9		4		9	
5	GSB 2	070	8	5		140		5		210	

Registros

0	RP1 (mod)	1	RP1 (arg)	2	RC1 (mod)	3	RC1 (arg)	4	RC2 (mod)	5	RC2 (arg)	6	RP2 (mod)
7	RP2 (arg)	8	V1 (mod)	9	V1 (arg)	.0	V2 (mod)	.1	V2 (arg)	.2	Δ (mod)	.3	Δ (arg)
.4	utiliz.	.5	utiliz.	.6		.7		.8		.9		I	

Etiquetas

A: CALCULA PESO CORRECTOR EN EL PLANO 1.

B: CALCULA PESO CORRECTOR EN EL PLANO 2.

0: CALCULA DETERMINANTE DEL SISTEMA.

1, 2, 3, 4, 5, 6 y 7.

Flags

Modo angular

DEG

Notacion