

## ORTOPOL: UN PROGRAMA EN QBASIC PARA EL ANÁLISIS DE TENDENCIAS Y EL AJUSTE DE FUNCIONES POR POLINOMIOS ORTOGONALES

Pedro Rodríguez-Miñón y Joan Torelló

U.N.E.D.

Se presenta un programa en QBASIC que ofrece el ajuste de funciones por polinomios ortogonales y, dentro de los procedimientos generales de los diseños experimentales, se analiza el estudio de las tendencias, su significación y proporción de varianza explicada, así como el valor de los coeficientes para contrastes ortogonales planificados de diferencias de medias, para el caso general de un diseño no equilibrado y con distinto espaciamiento entre los niveles de la variable independiente.

*ORTOPOL.BAS: A computer program for trend analysis and curves fitting by orthogonal polynomials.* A QBASIC program for curve fitting for orthogonal polynomials within experimental design is presented. It studies trend analysis, its significance and the proportion of explained variance and also the value of the coefficients for a priori orthogonal tests of differences between means. It is valid for the general case of unequally spaced and weighted treatment levels design.

En muchas situaciones experimentales, cuando queremos conocer el efecto que una variable independiente (V.I) o tratamiento tiene sobre la variable dependiente (V.D) aplicamos un análisis de varianza. Un valor de F significativo sólo nos indica que hay una variación general en la V.D en función de los distintos valores de la V.I, pero no sabemos nada más sobre los efectos reales de la V.I. Para poder conocer estos efectos reales planificamos análisis posteriores, donde lo primero que hay que distinguir es la naturaleza de la V.I. Si es

de tipo cuantitativo, podemos estar interesados en conocer la forma de la función que relaciona la V.I con la V.D, y/o en hacer comparaciones de medias entre los distintos niveles de la V.I. Para ello, disponemos del análisis de tendencias que consiste en un conjunto de procedimientos que nos permite evaluar la fortaleza de importantes fluctuaciones observadas en los datos y que nos dice la forma general que tendrá la función subyacente (Keppel, 1982) pero no nos proporciona dicha función resultante que nos permita hacer pronósticos o describir la relación entre las variables. Una alternativa a esta carencia la proporciona el ajuste de funciones por polinomios ortogonales que, a su objetivo final de encontrar la función que mejor se

---

Correspondencia: Pedro Rodríguez-Miñón  
Dpto. de Metodología de las CC. del Comportamiento  
Facultad de Psicología. U.N.E.D.  
Ciudad Universitaria. 28040 Madrid (Spain)

ajusta a los datos, añade el poder analizar, como el análisis de tendencias, cuál o cuáles de las tendencias son significativas.

El programa se ha desarrollado en QBASIC de MS-DOS. Puede funcionar en cualquier PC o compatible equipado con esta versión de BASIC o, precisando pequeñas modificaciones, con otras versiones. Los requerimientos de memoria son mínimos y dependen, en todo caso, de la extensión de los datos experimentales. Se puede obtener una copia del programa enviando un disco de 3.5" a los autores.

Previamente, señalaremos algunos aspectos fundamentales del ajuste de funciones por polinomios ortogonales para el caso general de diseños no equilibrados y donde la V.I no está igualmente espaciada.

#### Ajuste de funciones por polinomios ortogonales

Ajustar una función a unos datos por polinomios ortogonales consiste en encontrar la función polinómica de grado k:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \dots + \beta_k X^k$$

que se puede expresar como una combinación lineal de un conjunto de polinomios ortogonales  $P_i(X)$ :

$$Y' = b_0 P_0(X) + b_1 P_1(X) + b_2 P_2(X) + \dots + b_h P_h(X) + \dots + b_k P_k(X)$$

donde el término genérico,  $P_h(X)$ , es un polinomio en X de grado h con la siguiente expresión:

$$P_h(X) = C_{h0} + C_{h1} X + C_{h2} X^2 + \dots + C_{hh} X^h$$

y  $b_h$  ( $h = 0, 1, \dots, k$ ) es su coeficiente de regresión parcial.

Cuando en los diseños equilibrados la distancia entre los distintos niveles de tratamiento es constante, es posible utilizar

un conjunto predeterminado de coeficientes ortogonales  $C_{ij}$  que representan las distintas tendencias y que se encuentran recogidos en tablas que aparecen en la mayoría de los manuales de diseños experimentales (v.gr. Arnau, 1981; Keppel, 1982) y que también se utilizan para aplicar contrastes ortogonales planificados intergrupos. Pero cuando la distancia entre los distintos niveles de la V.I no es constante y/o no utilizamos un diseño equilibrado hay que proceder al cálculo de los mismos mediante la resolución de un sistema de h ecuaciones con h incógnitas. Este procedimiento, laborioso en cuanto a los cálculos que requiere, presenta la ventaja de que al ajustar un polinomio representativo de una tendencia de grado superior se mantienen los cálculos efectuados con anterioridad y los coeficientes de esta tendencia o polinomio de grado superior se obtienen en función de los coeficientes de los polinomios anteriores, sin tener que repetir todo el proceso. Diversos autores han desarrollado algoritmos para calcular estos coeficientes en estas condiciones (Gaito, 1965; Kirk, 1982). Por su parte Dunlap (1975), Possamai (1975), Coulombe (1985) y Weiss (1980, 1982) han diseñado distintos programas informáticos con el mismo fin. Wishart y Metakides (1953) presentaron un algoritmo o esquema de cálculo para el caso general -distancia entre niveles de tratamiento y tamaños de los grupos no necesariamente iguales- basado en el análisis de regresión múltiple que ajusta funciones por polinomios ortogonales, pero no indica cuáles son éstos.

Como característica distintiva, y también integradora de estos trabajos, ORTOPOL presenta los polinomios y coeficientes ortogonales y las funciones precisas -  $Y_1, Y_2, Y_3$  - que definen los mejores ajustes para cada una de las tendencias -lineal, cuadrática, cúbica- y se añade la representación gráfica de los correspondientes ajust-

tes. Si se desea imprimir estos gráficos, antes de entrar en el programa debe ejecutarse el comando GRAPHICS de MS-DOS.

### Significación de las tendencias

Un principio general es que si tenemos  $k$  niveles de la variable independiente, los datos pueden ajustarse por una función polinómica de grado  $k-1$  y la suma de cuadrados entre grupos (SCEG) puede descomponerse en tantos componentes de tendencia como grados de libertad asociados a los niveles de los tratamientos que estemos utilizando. Por tanto:  $SC(b_1) + SC(b_2) + \dots + SC(b_{k-1}) = SCEG$ . Una vez analizada la significación de la tendencia, el experimentador puede desear determinar su intensidad. En este programa se aplica el índice "omega cuadrado" descrito por Hays (1988; pág 307) que nos informa de la cantidad de varianza de la V.D explicada por la V.I. Pero además podemos utilizar este índice para evaluar la proporción de varianza explicada por cada una de las tendencias (lineal, cuadrática, cúbica, etc.) que no puede analizarse a partir de la significación del estadístico F.

### Características del programa

ORTOPOL, cuyo listado principal aparece en el Apéndice I, realiza el análisis de tendencias y ajusta funciones por polinomios ortogonales de primero, segundo y tercer grado a los datos experimentales para cualquier condición de tamaño de los grupos -equilibrados o no- y de espaciamento de la variable independiente -constante o no-. Tendencias de grado superior son difíciles de interpretar en el ámbito de la psicología y por esta razón no se han incluido.

Se trata de un programa flexible que permite la entrada directa de los datos experimentales sin necesidad de utilizar ningún tipo de plantilla o archivo de datos

previo, puesto que admite la introducción de datos sin agrupar y en orden aleatorio. Puede trabajar igualmente con valores de medias grupales, sin necesidad de introducir uno a uno los registros experimentales, si se dispone de datos ya elaborados.

ORTOPOL archiva y recupera los datos experimentales para su reutilización, junto con la tabla del ANOVA, los polinomios y coeficientes ortogonales y las funciones ajustadas. Asimismo ofrece una representación gráfica de los ajustes.

En el Apéndice II se reproduce parcialmente la salida del programa con un ejemplo extraído de Keppel (1982).

Inicialmente ha de determinarse si se recuperan los datos de un archivo, que se habrá creado en una sesión anterior. En caso afirmativo el programa requerirá el nombre del archivo y la vía de acceso.

Si la entrada de datos es manual se especificarán, en primer lugar, el número de grupos -niveles de tratamiento- y el número de ensayos experimentales. En caso de trabajar con medias de grupos se habrá de especificar un número de "ensayos" igual al número de grupos. Seguidamente, para cada "ensayo" o "grupo" se habrá de introducir su tamaño  $N$  -igual a 1 si se trata de ensayos individuales-, el valor del nivel de tratamiento  $X$ , y el valor de la variable dependiente  $Y$ .

A continuación el programa pide si los datos corresponden a medias grupales. En caso afirmativo requerirá la suma de cuadrados del error.

Seguidamente ofrece la tabla del ANOVA, en la que se recoge el nivel de significación del estadístico F y el índice "omega cuadrado" entre-grupos y de cada una de las tendencias.

Acto seguido, ORTOPOL presenta los polinomios y coeficientes ortogonales y las funciones precisas  $-Y_1, Y_2, Y_3-$  que definen los mejores ajustes para cada una de las tendencias -lineal, cuadrática, cúbica-.

En adelante -no se enseña en el ejemplo del Apéndice II -el programa representa gráficamente los correspondientes ajustes. Si se desea imprimir estos gráficos, antes de entrar en el programa debe ejecutarse el comando GRAPHICS de MS-DOS.

Finalmente, archiva en formato ASCII los datos experimentales, separados por comas, para ser utilizados en aplicaciones posteriores. En el mismo archivo guarda también la tabla del ANOVA, los polinomios y coeficientes ortogonales y las funciones ajustadas.

### Referencias

- Arnau, J. (1981). *Diseños experimentales en psicología y educación*. México: Trillas.
- Coulombe, D. (1985). Orthogonal polynomial coefficients and trend analysis for unequal intervals and unequal Ns: A microcomputer application. *Behavior Research Methods, Instruments & Computers*, 17 (3), 441-442.
- Dunlap, W. P. (1975). A FORTRAN IV program for trend analysis with unequal intervals and unequal Ns. *Behavior Research Methods & Instrumentation*, 7, 478- 479.
- Gaito, J. (1965). Unequal intervals and unequal n in trend analysis. *Psychological Bulletin*, 63, 125-127.
- Keppel, G. (1982). *Design Analysis. A Researcher's Handbook*. New Jersey: Prentice Hall.
- Kirk, R. E. (1982). *Experimental design: procedures for the behavioral sciences*. Belmont: Brooks/Cole.
- Possamai, C. A. (1975). CORAT: A FORTRAN subroutine to calculate orthogonal coefficients for trend analysis. *Behavior Research Methods & Instrumentation*, 7, 315-316.
- Weiss, D.J. (1980). ORCOPO: Orthogonal polynomial coefficient. *Behavior Research Methods & Instrumentation*, 12, 635.
- Weiss, D.J. (1982). REPCOP: Repeated-measures planned comparison and orthogonal polynomials. *Behavior Research Methods & Instrumentation*, 14,(1),44.
- Wishart, J. & Metakides, T. (1953). Orthogonal polynomial fitting. *Biometrika*, 40, 361-369.

Aceptado el 6 de mayo de 1996

**APÉNDICE I**

Listado principal de ORTOPOL

DECLARE SUB pausa ()

CLS : DEFINT K

REM \*\*ENTRADA DE DATOS TECLADO  
\*\*\*\*\*

20 INPUT "Number of groups"; j  
INPUT "Number of trials"; k  
DIM j(k), m(-1 TO k + 1), n(k), l(k), p(-1 TO k + 1), q(-1 TO k + 1)

FOR i = 1 TO k:PRINT "Trial/Group"; i;"  
: ";TAB(19);"N ==> " ;:LINE INPUT n\$: n(i)  
=VAL(n\$):nn = nn + n(i)

PRINT TAB(30); CHR\$(30); "Treatment X  
==> " ; : LINE INPUT m\$: m(i) = VAL(m\$): mn  
= mn + m(i)

PRINT TAB(55); CHR\$(30); "Dependent Y  
==> " ; : LINE INPUT l\$: l(i) = VAL(l\$): ln = ln  
+ l(i)

NEXT i

71 PRINT STRING\$(79, 205): CALL pausa

REM \*\*\*\*CALCULO COEFICIENTES DE  
REGRESION

\*\*\*\*\*

CLS : S00 = nn  
FOR i = 1 TO k: S10 = S10 + n(i) \* m(i)

S20 = S20 + n(i) \* m(i) ^ 2

S30 = S30 + n(i) \* m(i) ^ 3

S31 = S31 + n(i) \* m(i) ^ 4

S32 = S32 + n(i) \* m(i) ^ 5

S33 = S33 + n(i) \* m(i) ^ 6

Sy0 = Sy0 + n(i) \* l(i)

Sy1 = Sy1 + n(i) \* m(i) \* l(i)

Sy2 = Sy2 + n(i) \* m(i) ^ 2 \* l(i)

Sy3 = Sy3 + n(i) \* m(i) ^ 3 \* l(i)

Syy = Syy + n(i) \* l(i) ^ 2

NEXT i

b00 = S00 / S00

b10 = S10 / S00

b20 = S20 / S00

b30 = S30 / S00

by0 = Sy0 / S00

S11.0 = S20 - (b10 \* S10)

c10 = -b10 / S11.0

b11.0 = S11.0 / S11.0

S21.0 = S30 - (b10 \* S20)

b21.0 = S21.0 / S11.0

S31.0 = S31 - (b10 \* S30)

b31.0 = S31.0 / S11.0

Sy1.0 = Sy1 - (b10 \* Sy0)

by1.0 = Sy1.0 / S11.0

S22.01 = S31 + (-b20 \* S20) + (-b21.0 \* S21.0)

b20.1 = -b20 + (-b21.0 \* -b10)

c20.1 = b20.1 / S22.01

c21.0 = -b21.0 / S22.01

b22.01 = S22.01 / S22.01

S32.01 = S32 + (-b20 \* S30) + (-b21.0 \* S31.0)

b32.01 = S32.01 / S22.01

Sy2.01 = Sy2 + (-b20 \* Sy0) + (-b21.0 \* Sy1.0)

by2.01 = Sy2.01 / S22.01

S33.012 = S33 + (-b30 \* S30) + (-b31.0 \* S31.0)

+ (-b32.01 \* S32.01)

b30.12 = -b30 + (-b31.0 \* -b10) + (-b32.01 \*  
b20.1)

c30.12 = b30.12 / S33.012

b31.02 = -b31.0 + (-b32.01 \* -b21.0)

c31.02 = b31.02 / S33.012

c32.01 = -b32.01 / S33.012

b33.012 = S33.012 / S33.012

Sy3.012 = Sy3 + (-b30 \* Sy0) + (-b31.0 \* Sy1.0)

+ (-b32.01 \* Sy2.01)

by3.012 = Sy3.012 / S33.012

Syy.0123 = (by0 \* Sy0) + (by1.0 \* Sy1.0) +

(by2.01 \* Sy2.01) + (by3.012 \* Sy3.012)

REM \*\*\*\* ANALISIS DE VARIANZA

\*\*\*\*\*

991 INPUT "Are data group means (y/n)"; x\$

SELECT CASE x\$

CASE "y": GOTO 993: CASE "Y": GOTO 993

CASE "n": GOTO 992: CASE "N": GOTO 992

CASE ELSE: GOTO 991

END SELECT

992 DFt = k - 1

SSt = Syy - (Sy0 ^ 2 / S00)

SSe = Syy - Syy.0123

GOTO 995

993 DFt = nn - 1

INPUT "Residual Sum of Squares"; z: PRINT

SSe = z

SSt = Syy - (Sy0 ^ 2 / S00) + SSe

995 SStr = SSt - SSe

DFtr = j - 1

DFe = nn - j

MSe = SSe / DFe

MStr = SStr / DFtr

Ftr = MStr / MSe

SSconst = by0 \* Sy0

Fconst = SSconst / MSe

SSLin = by1.0 \* Sy1.0

Flin = SSLin / MSe

SSquad = by2.01 \* Sy2.01

```

Fquad = SSquad / MSe
SScub = by3.012 * Sy3.012
Fcub = SScub / MSe
GOTO 1000
996 Wtr = (SStr - (j - 1) * MSe) / (SSt + MSe)
Wtr1 = (SSlin - MSe) / (SSt + MSe)
Wtr2 = (SSquad - MSe) / (SSt + MSe)
Wtr3 = (SScub - MSe) / (SSt + MSe)
PRINT TAB(50); "Press any key to continue"
CLS : PRINT TAB(32); "ANOVA TABLE":
PRINT STRING$(73, 205)
USS = "#####.### #.### #.###"
PRINT "SOURCE ";TAB (16); "SS";TAB (26);
"DF";TAB (37); "MS";TAB (48); "F";TAB (58);
"P(F)";TAB (67); "Omega 2":
PRINT STRING$(73, 45)
PRINT "BETWEEN "; : PRINT USING USS;
SStr; DFtr; MStr; Ftr; PFtr; Wtr:
PRINT "LIN. Y1"; : PRINT USING USS; SS-
lin; 1; SSlin; Flin; Pflin; Wtr1:
PRINT "QUAD. Y2"; : PRINT USING USS;
SSquad; 1; SSquad; Fquad; PFquad; Wtr2:
PRINT "CUBIC Y3"; : PRINT USING USS;
SScub; 1; SScub; Fcub; PFcub; Wtr3:
PRINT "WITHIN "; : PRINT USING USS;
SSE; DFe; MSe: PRINT STRING$(73, 45)
PRINT "TOTAL "; : PRINT USING USS; SSt;
DFt
GOTO 1006
1000 IF Ftr < 1 THEN SV = nn - j: T = j - 1: z =
1 / Ftr ELSE SV = j - 1: T = nn - j: z = Ftr
W = 2 / 9 / SV: 1 = 2 / 9 / T: Y = ABS((1 - 1) * z
^ (1 / 3) - 1 + W) / SQR(1 * z ^ (2 / 3) + W)
IF T < 4 THEN Y = Y * (1 + .08 * Y ^ 4 / T ^ 3)
PFtr = .5 / (1 + Y * (.196854 + Y * (.115194 + Y
* (.000344 + Y * .019527)))) ^ 4
IF Ftr < 1 THEN PFtr = 1 - PFtr
IF Flin < 1 THEN SV = nn - j: T = 1: z = 1 / Flin
ELSE SV = 1: T = nn - j: z = Flin
W = 2 / 9 / SV: 1 = 2 / 9 / T: Y = ABS((1 - 1) * z
^ (1 / 3) - 1 + W) / SQR(1 * z ^ (2 / 3) + W)
IF T < 4 THEN Y = Y * (1 + .08 * Y ^ 4 / T ^ 3)
Pflin = .5 / (1 + Y * (.196854 + Y * (.115194 +
Y * (.000344 + Y * .019527)))) ^ 4
IF Flin < 1 THEN Pflin = 1 - Pflin
IF Fquad < 1 THEN SV = nn - j: T = 1: z = 1 /
Fquad ELSE SV = 1: T = nn - j: z = Fquad
W = 2 / 9 / SV: 1 = 2 / 9 / T: Y = ABS((1 - 1) * z
^ (1 / 3) - 1 + W) / SQR(1 * z ^ (2 / 3) + W)
IF T < 4 THEN Y = Y * (1 + .08 * Y ^ 4 / T ^ 3)
PFquad = .5 / (1 + Y * (.196854 + Y * (.115194
+ Y * (.000344 + Y * .019527)))) ^ 4
IF Fquad < 1 THEN PFquad = 1 - PFquad

```

```

IF Fcub < 1 THEN SV = nn - j: T = 1: z = 1 /
Fcub ELSE SV = 1: T = nn - j: z = Fcub
W = 2 / 9 / SV: 1 = 2 / 9 / T: Y = ABS((1 - 1) * z
^ (1 / 3) - 1 + W) / SQR(1 * z ^ (2 / 3) + W)
IF T < 4 THEN Y = Y * (1 + .08 * Y ^ 4 / T ^ 3)
PFcub = .5 / (1 + Y * (.196854 + Y * (.115194 +
Y * (.000344 + Y * .019527)))) ^ 4
IF Fcub < 1 THEN PFcub = 1 - PFcub
1001 GOTO 996
1006 PRINT STRING$(73, 205): CALL pausa

```

```

REM ***** POLINOMIOS Y
COEFICIENTES ORTOGONALES
*****
cc00 = ABS(b00): cc10 = ABS(b10): cc20 =
ABS(b20.1): cc21 = ABS(b21.0): cc32 =
ABS(b32.01): cc30 = ABS(b30.12): cc31 =
ABS(b31.02)
FOR i = 1 TO k: p(i) = m(i): NEXT i
z = k
WHILE (z >= 2)
FOR i = 2 TO z
IF p(i - 1) > p(i) THEN SWAP p(i - 1), p(i)
NEXT i
z = z - 1
WEND
ut$ = "#####.###"
CLS : PRINT TAB(21); "ORTHOGONAL
POLYNOMIAL AND COEFFICIENTS"
PRINT TAB(37); "Linear": PRINT
a = S00 * cc10 + S10: IF .01 > a AND -.01 < a
THEN PRINT TAB(27); "P1(X) = "; cc10; "+ ";
"X": PRINT
IF .01 > a AND -.01 < a THEN
FOR i = 1 TO k
IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$, TAB(43);
cc10 + q(i)
NEXT i
END IF
b = S00 * (-cc10) + S10: IF .01 > b AND -.01 <
b THEN PRINT TAB(27); "P1(X) = "; -cc10; "+
"; "X": PRINT
IF .01 > b AND -.01 < b THEN
FOR i = 1 TO k
IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$, TAB(43);
-cc10 + q(i)
NEXT i
END IF
CALL pausa
PRINT TAB(37); "Quadratic": PRINT
aa = S10 * cc20 + S20 * cc21 + S30: IF .01 > aa

```

```

AND -.01 < aa THEN PRINT TAB(24); "P2(X)
= "; cc20; " + "; cc21; "X"; " + "; "X^2": PRINT
IF .01 > aa AND -.01 < aa THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
cc20 + cc21 * q(i) + q(i) ^ 2
NEXT i
END IF
bb = S10 * cc20 + S20 * (-cc21) + S30: IF .01 >
bb AND -.01 < bb THEN PRINT TAB(24);
"P2(X) = "; cc20; " - "; cc21; "X"; " + "; "X^2":
PRINT
IF .01 > bb AND -.01 < bb THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
cc20 - cc21 * q(i) + q(i) ^ 2
NEXT i
END IF
cc = S10 * (-cc20) + S20 * cc21 + S30: IF .01 >
cc AND -.01 < cc THEN PRINT TAB(24);
"P2(X) = "; -cc20; " + "; cc21; "X"; " + ";
"X^2": PRINT
IF .01 > cc AND -.01 < cc THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
-cc20 + cc21 * q(i) + q(i) ^ 2
NEXT i
END IF
dd = S10 * (-cc20) + S20 * (-cc21) + S30: IF .01
> dd AND -.01 < dd THEN PRINT TAB(24);
"P2(X) = "; -cc20; " - "; cc21; "X"; " + "; "X^2":
PRINT
IF .01 > dd AND -.01 < dd THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
-cc20 - cc21 * q(i) + q(i) ^ 2
NEXT i
END IF
CALL pausa
PRINT TAB(37); "Cubic": PRINT
aaa = S10 * cc30 + S20 * cc31 + S30 * cc32 +
S31: IF .01 > aaa AND -.01 < aaa THEN PRINT
TAB(20); "P3(X)= "; cc30; " + "; cc31; "X"; " +
"; cc32; "X^2"; " + "; "X^3": PRINT
IF .01 > aaa AND -.01 < aaa THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
cc30 + cc31 * q(i) + cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
fff = S10 * cc30 + S20 * cc31 + S30 * (-cc32) +
S31: IF .01 > fff AND -.01 < fff THEN PRINT
TAB(20); "P3(X) = "; cc30; " + "; cc31; "X"; " -
"; cc32; "X^2"; " + "; "X^3": PRINT
IF .01 > fff AND -.01 < fff THEN
FOR i = 1 TO k

```

```

NEXT i
END IF
bbb = S10 * (-cc30) + S20 * cc31 + S30 * cc32
+ S31: IF .01 > bbb AND -.01 < bbb THEN
PRINT TAB(20); "P3(X)= "; -cc30; " + "; cc31;
"X"; " + "; cc32; "X^2"; " + "; "X^3": PRINT
IF .01 > bbb AND -.01 < bbb THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
-cc30 + cc31 * q(i) + cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
ccc = S10 * (-cc30) + S20 * (-cc31) + S30 * cc32
+ S31: IF .01 > ccc AND -.01 < ccc THEN
PRINT TAB(20); "P3(X) = "; -cc30; " - "; cc31;
"X"; " + "; cc32; "X^2"; " + "; "X^3": PRINT
IF .01 > ccc AND -.01 < ccc THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
-cc30 - cc31 * q(i) + cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
ddd = S10 * (-cc30) + S20 * cc31 + S30 * (-
cc32) + S31: IF .01 > ddd AND -.01 < ddd
THEN PRINT TAB(20); "P3(X) = "; -cc30; " +
"; cc31; "X"; " - "; cc32; "X^2"; " + "; "X^3":
PRINT
IF .01 > ddd AND -.01 < ddd THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
-cc30 + cc31 * q(i) - cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
eee = S10 * cc30 + S20 * (-cc31) + S30 * (-cc32)
+ S31: IF .01 > eee AND -.01 < eee THEN
PRINT TAB(20); "P3(X) = "; cc30; " - "; cc31;
"X"; " - "; cc32; "X^2"; " + "; "X^3": PRINT
IF .01 > eee AND -.01 < eee THEN
FOR i = 1 TO k
  IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C("; q(i); ")"; USING ut$; TAB(43);
cc30 - cc31 * q(i) - cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
fff = S10 * cc30 + S20 * cc31 + S30 * (-cc32) +
S31: IF .01 > fff AND -.01 < fff THEN PRINT
TAB(20); "P3(X) = "; cc30; " + "; cc31; "X"; " -
"; cc32; "X^2"; " + "; "X^3": PRINT
IF .01 > fff AND -.01 < fff THEN
FOR i = 1 TO k

```

```

IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C(“; q(i); “)”; USING ut$; TAB(43);
cc30 + cc31 * q(i) - cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
ggg = S10 * cc30 + S20 * (-cc31) + S30 * cc32
+ S31: IF .01 > ggg AND -.01 < ggg THEN
PRINT TAB(20); "P3(X) = “; cc30; “ - “; cc31;
“X”; “ + “; cc32; “X^2”; “ + “; “X^3”: PRINT
IF .01 > ggg AND -.01 < ggg THEN
FOR i = 1 TO k
IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C(“; q(i); “)”; USING ut$; TAB(43);
cc30 - cc31 * q(i) + cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
hhh = S10 * (-cc30) + S20 * (-cc31) + S30 * (-
cc32) + S31: IF .01 > hhh AND -.01 < hhh
THEN PRINT TAB(20);
“P3(X) = “; -cc30; “ - “; cc31; “X”; “ - “; cc32;
“X^2”; “ + “; “X^3”: PRINT
IF .01 > hhh AND -.01 < hhh THEN
FOR i = 1 TO k
IF p(i) <> p(i + 1) THEN q(i) = p(i): PRINT
TAB(27); "C(“; q(i); “)”; USING ut$; TAB(43);
-cc30 - cc31 * q(i) - cc32 * q(i) ^ 2 + q(i) ^ 3
NEXT i
END IF
CALL pausa

```

```

REM *****FUNCIONES POLINOMICAS
*****
CLS : by0.1 = by0 - (-b10 * -by1.0)
by0.12 = by0.1 - (b20.1 * -by2.01)
by0.123 = by0.12 - (b30.12 * -by3.012)
by1.02 = by1.0 - (-b21.0 * -by2.01)
by1.023 = by1.02 - (b31.02 * -by3.012)
by2.013 = by2.01 - (-b32.01 * -by3.012)
PRINT TAB(24); "ORTHOGONAL POLYNO-
MIAL FUNCTIONS": PRINT STRING$(79,
205)
PRINT "CONST. Y0= “; by0
PRINT "LIN. Y1= “; by0.1; “+ “; by1.0; “X”
PRINT "QUAD. Y2= “; by0.12; “+ “; by1.02;
“X”; “ + “; by2.01; “X^2”
PRINT "CUBIC Y3= “; by0.123; “+ “; by1.023;
“X”; “ + “; by2.013; “X^2”; “ + “; by3.012;
“X^3”
PRINT STRING$(79, 205): CALL pausa

```

```

REM *****REPRESENTACION
GRAFICA DE LAS FUNCIONES
*****

```

```

CLS : max = m(1): min = m(1)
FOR i = 1 TO k
IF m(i) > max THEN max = m(i)
IF m(i) < min THEN min = m(i)
NEXT i
lmax = l(1): lmin = l(1)
FOR i = 1 TO k
IF l(i) > lmax THEN lmax = l(i)
IF l(i) < lmin THEN lmin = l(i)
NEXT i
SCREEN 10
VIEW (20, 70)-(600, 340)
WINDOW (min, lmin)-(max, lmax)
PRINT TAB(32); "LINEAR TREND: Y1":
PRINT TAB(12); "Origin (“; min; “,”; lmin; “)
X axis”; min; “to”; max; “ Y axis”; lmin; “to”;
lmax
LINE (min, lmin)-(max, lmin)
LINE (min, lmin)-(min, lmax)
r = (max - min) / (10 * j)
FOR r = min TO max STEP r: PSET (r, by0.1 +
by1.0 * r)
NEXT r
FOR i = 1 TO k: PSET (m(i), l(i)), 2
NEXT i
CALL pausa
SCREEN 0
CLS : SCREEN 10
VIEW (20, 70)-(600, 340)
WINDOW (min, lmin)-(max, lmax)
PRINT TAB(32); "QUAD TREND: Y2":
PRINT TAB(12); "Origin (“; min; “,”; lmin; “)
X axis”; min; “to”; max; “ Y axis”; lmin; “to”;
lmax
LINE (min, lmin)-(max, lmin)
LINE (min, lmin)-(min, lmax)
r = (max - min) / (10 * j)
FOR r = min TO max STEP r: PSET (r, by0.12 +
by1.02 * r + by2.01 * r ^ 2)
NEXT r
FOR i = 1 TO k: PSET (m(i), l(i)), 2
NEXT i
CALL pausa
SCREEN 0
CLS : SCREEN 10
VIEW (20, 70)-(600, 340)
WINDOW (min, lmin)-(max, lmax)
PRINT TAB(32); "CUBIC TREND: Y3":
PRINT TAB(12); "Origin (“; min; “,”; lmin; “)
X axis”; min; “to”; max; “ Y axis”; lmin; “to”;
lmax
LINE (min, lmin)-(max, lmin)
LINE (min, lmin)-(min, lmax)

```



```

r = (max - min) / (10 * j)
FOR r = min TO max STEP r: PSET (r, by0.123
+ by1.023 * r + by2.013 * r ^ 2 + by3.012 * r ^
3)
NEXT r
FOR i = 1 TO k: PSET (m(i), l(i)), 2
NEXT i
CALL pausa
CLS : SCREEN 0
END
DEF SNG K
SUB pausa
BEEP
PRINT
PRINT TAB(55); "press any key to continue"
DO
LOOP UNTIL INKEY$ <> ""
PRINT
END SUB

```

**APÉNDICE II**

**Ejemplo parcial del funcionamiento de ORTOPOL**

Get data from a file (y/n)? n

Number of groups? 9  
 Number of trials? 14

Trial/Group 1 : N==> 1	Treatment X==> 0	Dependent Y==> 0
Trial/Group 2 : N==> 1	Treatment X==> 1	Dependent Y==> 2
Trial/Group 3 : N==> 1	Treatment X==> 1	Dependent Y==> 2
Trial/Group 4 : N==> 1	Treatment X==> 3	Dependent Y==> 5
Trial/Group 5 : N==> 1	Treatment X==> 4	Dependent Y==> 6
Trial/Group 6 : N==> 1	Treatment X==> 7	Dependent Y==> 9
Trial/Group 7 : N==> 1	Treatment X==> 7	Dependent Y==> 9
Trial/Group 8 : N==> 1	Treatment X==> 8	Dependent Y==> 8
Trial/Group 9 : N==> 1	Treatment X==> 8	Dependent Y==> 8
Trial/Group 10 : N==> 1	Treatment X==> 8	Dependent Y==> 8
Trial/Group 11 : N==> 1	Treatment X==> 10	Dependent Y==> 7
Trial/Group 12 : N==> 1	Treatment X==> 11	Dependent Y==> 5
Trial/Group 13 : N==> 1	Treatment X==> 11	Dependent Y==> 5
Trial/Group 14 : N==> 1	Treatment X==> 12	Dependent Y==> 4

Press any key to continue

Are data group means (y/n)? n

Press any key to continue

**ANOVA TABLE**

SOURCE	SS	DF	MS	F	P(F)	Omega 2
BETWEEN	101.510	8	12.689	33.062	0.0016	0.9482
LIN. Y1	31.792	1	31.792	82.837	0.0007	0.3025
QUAD. Y2	69.230	1	69.230	180.386	0.0003	0.6632
CUBIC Y3	0.488	1	0.488	1.270	0.3115	0.0010
WITHIN	1.919	5	0.384			
TOTAL	103.429	13				

Press any key to continue

ORTHOGONAL POLYNOMIAL AND COEFFICIENTS

	Linear	
P1(X) = -6.5 + X		
C( 0 )		-6.500
C( 1 )		-5.500
C( 3 )		-3.500
C( 4 )		-2.500
C( 7 )		0.500
C( 8 )		1.500
C( 10 )		3.500
C( 11 )		4.500
C( 12 )		5.500
	Quadratic	
P2(X) = 19.02938		- 11.75177 X + X^2
C( 0 )		19.029
C( 1 )		8.278
C( 3 )		-7.226
C( 4 )		-11.978
C( 7 )		-14.233
C( 8 )		-10.985
C( 10 )		1.512
C( 11 )		10.760
C( 12 )		22.008
	Cubic	
P3(X) = -63.7631 + 89.4772 X		- 18.62359 X^2 + X^3
C( 0 )		-63.763
C( 1 )		8.091
C( 3 )		64.056
C( 4 )		60.168
C( 7 )		-6.979
C( 8 )		-27.855
C( 10 )		-31.350
C( 11 )		-1.968
C( 12 )		56.167

Press any key to continue

ORTHOGONAL POLYNOMIAL FUNCTIONS

---

CONST. Y0= 5.571429

LIN. Y1= 3.051334 + .3877068 X

QUAD. Y2= -0.3396495 + 2.481841 X + -0.1781973 X^2

CUBIC Y3= -1.223215E-02 + 2.022384 X + -8.256696E-02 X^2 + -5.134903E-03 X^3

---

Press any key to continue