Science & Technology
Facilities Council

PE04

PACKAGE SPECIFICATION HSL ARCHIVE

1 SUMMARY

Given a function $y(x)$ in the range $u \le x \le v$ **finds a rational approximation** of the form

$$R_q^p(z) = \frac{a_0T_0(z)+a_1T_1(z)+...+a_pT_p(z)}{T_0(z)+b_1T_1(z)+...+b_qT_q(z)}$$

where $T_k(z)$ is the Chebyshev polynomial of order k and $-1 \le z = \frac{2x-u-v}{v-u} \le 1$ and $0 \le p \le 10$, $0 \le q \le 10$.

The approximation found is such that $R_q^p(z_k) = y(\frac{1}{2}\{z_k(v-u)+v+u\})$ k=0, 1,..., p+q where the z_k are the zeros of $T_{p+q+1}(z)$.

The subroutine returns a guide to how close the approximation is to the *best approximation* in the minimax sense. Also returned are the coefficients of the equivalent form

$$\tilde{R}_q^p(x) = \frac{\alpha_0+\alpha_1x+...+\alpha_px^p}{\beta_0+\beta_1x+...+\beta_qx^q}$$

The user must provide a function subroutine to evaluate $y(x)$.

ATTRIBUTES — **Version:** 1.0.0. **Types:** PE04A; PE04AD. **Calls:** MA21 and FUNCT (a user subroutine). **Original date:** August 1963. **Origin:** A.R.Curtis, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list

The single precision version

```
CALL PE04A(K,L,IP,IQ,U,V,A,B,ALPHA,BETA,E1,E2,EP1,EP2)
```

The double precision version

```
CALL PE04AD(K,L,IP,IQ,U,V,A,B,ALPHA,BETA,E1,E2,EP1,EP2)
```

K is an INTEGER which must be set by the user to control the printing of results. There are basically four groups of numbers which may be printed:

(i) the arrays A and B,

(ii) the arrays ALPHA and BETA,

(iii) E1, E2, EP1 and EP2,

(iv) a table of x_i^*, y_i^*, $R_q^p(x_i^*)$, δy_i^* and $\delta y_i^*/y_i^*$ (see E1, E2, etc.).

The control variable K must lie in the range 1 to 8 and has the following effect:

1 no printing is done,

2 (i) and (iii) are printed,

3 (ii) and (iii) are printed,

4 (i), (ii) and (iii) are printed,

5 (iii) and (iv) are printed,

6 (i), (iii) and (iv) are printed,

7 (ii), (iii) and (iv) are printed,

8 (i), (ii), (iii) and (iv) are printed.

`L` is an `INTEGER` variable which must be set to a case number of the user's own choice. It is not used by the subroutine but is printed on any output and also passed as an argument to the user supplied subroutine described in section §2.2.

`IP` is an `INTEGER` variable which must be set by the user to p the degree of the numerator of the rational function. **Restriction:** $0 \le p \le 10$.

`IQ` is an `INTEGER` variable which must be set by the user to q the degree of the denominator of the rational function. **Restriction:** $0 \le q \le 10$.

`U` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which must be set by the user to u the lower limit of the approximation range.

`V` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which must be set by the user to v the upper limit of the approximation range.

`A` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least p+1 in which the subroutine will return the calculated coefficients of the numerator of $R_q^p(x)$, i.e. the subroutine will set `A(k+1)` = a_k, k=0, 1, 2,..., p.

`B` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least q+1 in which the subroutine will return the calculated coefficients of the denominator of $R_q^p(x)$, i.e. the subroutine will set `B(k+1)` = b_k, k=0, 1, 2,..., q.

`ALPHA` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least p+1 in which the subroutine will return the calculated coefficients of the numerator of $\tilde{R}_q^p(x)$, i.e. the subroutine will set `ALPHA(k+1)` = α_k, k=0, 1, 2,..., p.

`BETA` is a `REAL` (`DOUBLE PRECISION` in the D version) array of length at least q+1 in which the subroutine will return the calculated coefficients of the denominator of $\tilde{R}_q^p(x)$, i.e. the subroutine will set `BETA(k+1)` = β_k, k=0, 1, 2,..., p.

`E1`, `E2`, `EP1` and `EP2` are `REAL` (`DOUBLE PRECISION` in the D version) variables which are set by the subroutine to give information about the goodness of fit. They are obtained by comparing the function and the fit on a set of points x_i^*, i=1, 2,..., q+p+2 which include u and v and are separated by the q+p+1 points used in the fitting. Let

$$\left.\begin{array}{l} \delta y_i^* = R_q^p(x_i^*) - y(x_i^*) \\ y_i^* = y(x_i^*) \end{array}\right\} \qquad (4)$$

then

$$\left.\begin{array}{l} \mathtt{E1} = \max\limits_i |\delta y_i^*| \\ \mathtt{E2} = \max\limits_i |\delta y_i^* / y_i^*| \\ \mathtt{EP1} = \mathtt{E1} / \min\limits_i |\delta y_i^*| \\ \mathtt{EP2} = \mathtt{E2} / \min\limits_i |\delta y_i^* / y_i^*| \end{array}\right\} \qquad (5)$$

The value of `EP1` (or `EP2`) is a guide to the maximum reduction that could be expected in the absolute (or relative) error by varying the coefficients in (2), so that the fit is close to 'best possible' if the appropriate number is close to unity. `E1` (or `E2`) gives the actual maximum error.

2.2 The function subroutine

The user must provide a function subroutine called `FUNCT` which given a value of x returns the value of $y(x)$.

The single precision version

```
REAL FUNCTION FUNCT(X,L)
```

The double precision version

```
DOUBLE PRECISION FUNCTION FUNCT(X,L)
```

`X` is a `REAL` (`DOUBLE PRECISION` in the D version) variable which will have been set by `PE04` to the value of x. The user is expected to use it to calculate the value of the function $y(x)$.

`L` is an `INTEGER` variable which is passed unchanged from the user's call to `PE04` and can be used as a case number to generate different functions $y(x)$ on different calls to `PE04` (see argument `L` in §2.1).

`FUNCT` is the `REAL` (`DOUBLE PRECISION` in the D version) function variable which must be set by the user to the value of $y(x)$ before returning to `PE04`.

2.3 Error returns and diagnostics

The denominator

$$\sum_{k=0}^{p} b_k T_k(z)$$

of $R_q^p(z)$ is tested to ensure that it maintains a constant sign throughout the range; if not, the fitted function will have a singularity. If this occurs a diagnostic is printed, together with the output (i), and the argument `EP1` is set to –1.0 before returning.

3 GENERAL INFORMATION

Workspace: None.

Use of common: None.

Other routines called directly: `MA21A/AD` and a user supplied function subroutine which must be called `FUNCT` in both single and double versions.

Input/output: Printed output is controlled by argument `K` and is output on Fortran unit 6.

Restrictions: $0 \le p \le 10$, $0 \le q \le 10$. For high order fits to analytic functions the error is often dominated by rounding errors, `E2` being of order 10^{-12}. `PE04` is not suitable for fitting empirical functions, since only q function values are used and no smoothing is done (see the method section).

4 METHOD

The equations

$$\sum_{k=1}^{n} \cos t\theta_k \left(\sum_{r=0}^{p} a_r \cos r\theta_k - y(x_k) \sum_{s=0}^{q} b_s \cos s\theta_k \right) = 0 \qquad (6)$$

where $x_k = \tfrac{1}{2}(u + v + (v–u)\cos\theta_k)$ and $\theta_k = (k–\tfrac{1}{2})\pi/n$ are set up for t=0, 1, 2,..., p+q. The last q of them which do not involve the a_r, are solved for b_1, b_2,..., b_q taking $b_0 = 1$. The first p+1 equations then give a_0, a_1,..., a_p in terms of the (b_i)s. A direct conversion algorithm then converts the numerator and denominator to the form of polynomials in x, which are normalized to give $\beta_0 = 1$. The errors are computed at the points

$$x_k^* = \tfrac{1}{2}(u + v + (v–u)\cos\left\{ \frac{(k–1)\pi}{n} \right\} \qquad (7)$$

for k=1, 2,..., n+1.

This method makes the approximation exact at the points x_k, k=1, 2,..., n, which are the zeros of $T_n(z)$.

Some remarks

The subroutine may be used as a polynomial fitting subroutine, by setting q=0. In general, however, fits with $p \approx q$ tend to be better than those, with the same value of n, having very different degrees in numerator and denominator. For example, the following results were obtained with $y(x) = e^x$, $u = -1$ and $v = 1$.

p	q	E2	EP2
0	4	0.00125	5.29
1	3	0.000204	2.30
2	2	0.000089	1.06
3	1	0.000204	2.30
4	0	0.00125	5.30

Clearly the (2,2) fit is best.