# ALGORITHM 592
# A FORTRAN Subroutine for Computing the Optimal Estimate of $f(x)$

P. W. GAFFNEY
Oak Ridge National Laboratory

Categories and Subject Descriptors: G.1.1. [**Numerical Analysis**]: Interpolation—*interpolation formulas*; *spline and piecewise polynomial interpolation*, G.1.2. [**Numerical Analysis**]: Approximation—*nonlinear approximation*; G.m. [**Mathematics of Computing**]: Miscellaneous—*FORTRAN*
General Terms. Algorithms
Additional Key Words and Phrases: Optimal estimation, optimal interpolation, perfect splines

## 1. INTRODUCTION

In this paper we present a FORTRAN subroutine to compute the solution of the following problem.

Given values of an unknown function $f$ at $n$ distinct points, $x_1 < x_2 < \cdots < x_{n-1} < x_n$, and given an integer $k$, $1 \le k \le n$, and a finite bound on the $k$th derivative of $f$,

$$\| f^{(k)} \|_\infty \equiv \max | f^{(k)}(x) | \le L < \infty, \qquad x_1 \le x \le x_n,$$

determine the range of possible values of $f(\alpha)$ (and hence the optimal estimate of $f(\alpha)$), where $\alpha$ is any point in the interval $x_1 \le x \le x_n$.

Gaffney [5] and Gaffney and Powell [6] have solved this problem. They proved that the *closest possible* bounds on $f(\alpha)$ are given by the interval

$$\min[u(\alpha), l(\alpha)] \le f(\alpha) \le \max[u(\alpha), l(\alpha)], \tag{1.1}$$

where the quantities $u(\alpha)$ and $l(\alpha)$ are the values at $x = \alpha$ of two perfect splines of degree $k$ which pass through the given function values. A method for computing the range (1.1) is described in a companion paper by Gaffney [7]. Therefore, the purpose of this paper is to present a FORTRAN subroutine for computing the values $u(\alpha)$, $l(\alpha)$, and the estimate of $f(\alpha)$ whose error has the smallest possible bound, that is, the quantity

$$\Omega(\alpha, L) = \frac{(u(\alpha) + l(\alpha))}{2}. \tag{1.2}$$

Received 20 January 1982; revised 8 July 1982; accepted 3 September 1982
Oak Ridge National Laboratory is operated by Union Carbide Corporation under Contract W-7405-eng-26 with the U.S. Department of Energy.
Author's address: Computer Sciences, Oak Ridge National Laboratory, Building 9104-2, P.O. Box Y, Oak Ridge, TN 37830.
1983 ACM 0098-3500/83/0300-0098 $00.00

The name of this subroutine is **RANGE**.

It is important that prospective users of **RANGE** are aware of the amount of computation involved in computing the numbers $u(\alpha)$ and $l(\alpha)$. Therefore, in Section 2 we give a brief description of the method used by **RANGE**. In Section 3 we present a sample program for a situation where **RANGE** may be used. In Section 4 we discuss some aspects of practical approximation that we believe may be useful to prospective users of **RANGE**. We recommend that these users should read this section, in particular the conclusions at the end of the section, *before* incorporating subroutine **RANGE** in a FORTRAN program. In Section 5 we describe the standards that subroutine **RANGE** adheres to, and in Section 6 we present a flowchart that describes the way in which **RANGE** should be called. At the end of the paper we present the FORTRAN listing of subroutine **RANGE**.

In addition to the work cited in this paper, a number of other authors have also considered optimal approximation schemes. Some of this work is described in the book by Micchelli and Rivlin [9]. For further discussion on optimal interpolation, with particular reference to the role played by natural spline interpolation, we refer the interested reader to the thorough exposition given by Powell [11].

## 2. METHOD OF COMPUTATION

In this section we give a brief description of the algorithm used by **RANGE**. To do this we first recall, from [7], the solution of the optimal estimation problem and we review the properties of the functions $u$ and $l$ that provide the bounds (1.1).

Given values $f_1, \ldots, f_n$ of a function $f$ at the points

$$x_1 < x_2 < \cdots < x_{n-1} < x_n,$$

and given that the inequality

$$\max | f^{(k)}(x) | \le L < \infty, \qquad 1 \le k \le n, \quad x_1 \le x \le x_n,$$

is satisfied, where the value of $L$ is greater than the least value of

$$\max | f^{(k)}(x) |, \qquad x_1 \le x \le x_n$$

that is consistent with the function values $f_1, \ldots, f_n$, then the *closest possible* bounds on $f(x)$, for *any* $x$ in the range $x_1 \le x \le x_n$, are given by the inequalities

$$\min[u(x), l(x)] \le f(x) \le \max[u(x), l(x)]. \tag{2.1}$$

The functions $u$ and $l$ in expression (2.1) are perfect splines of degree $k$; they each have $n - k$ *knots*, and they each satisfy the interpolation conditions

$$u(x_i) = l(x_i) = f(x_i), \qquad i = 1, \ldots, n. \tag{2.2}$$

Furthermore, the $k$th derivative of $u$ satisfies the equation

$$u^{(k)}(x) = \begin{cases} L & x_1 \le x < \eta_1 \\ (-1)^i L & \eta_i \le x < \eta_{i+1}, \qquad i = 1, \ldots, n-k-1 \\ (-1)^{n-k} L & \eta_{n-k} \le x \le x_n \end{cases} \tag{2.3}$$

and the $k$th derivative of $l$ satisfies the equation

$$l^{(k)}(x) = \begin{cases} -L & x_1 \le x < \xi_1 \\ (-1)^{i+1} L & \xi_i \le x < \xi_{i+1}, \qquad i = 1, \ldots, n-k-1 \\ (-1)^{n-k+1} L & \xi_{n-k} \le x \le x_n. \end{cases} \tag{2.4}$$

We calculate the knots $\{\eta_i\}$ and $\{\xi_i\}$ by solving the systems of equations

$$\sum_{j=0}^{n-k} (-1)^j \int_{\eta_j}^{\eta_{j+1}} M_{k,i}(x)\,dx - L^{-1}(k-1)!f(x_i, \ldots, x_{i+k}) = 0, \qquad i = 1, \ldots, n-k \quad (2.5)$$

and

$$\sum_{j=0}^{n-k} (-1)^{j+1} \int_{\xi_j}^{\xi_{j+1}} M_{k,i}(x)\,dx - L^{-1}(k-1)!f(x_i, \ldots, x_{i+k}) = 0, \qquad i = 1, \ldots, n-k \quad (2.6)$$

where $\eta_0 = \xi_0 = x_1$, $\eta_{n-k+1} = \xi_{n-k+1} = x_n$, $M_{k,i}(x)$ is a B-spline of degree $k-1$ with knots $x_i, \ldots, x_{i+k}$, and $f(x_i, \ldots, x_{i+k})$ is the $k$th divided difference of $f$ based on the points $x_i, \ldots, x_{i+k}$.

When $k = 1$, eqs. (2.5) and (2.6) are linear. In this case it is straightforward to show that the knots $\{\eta_i\}$ and $\{\xi_i\}$ have the values

$$\eta_i = \frac{(-1)^{i+1}}{2L}(f(x_{i+1}) - f(x_i)) + \frac{(x_i + x_{i+1})}{2}, \qquad i = 1, \ldots, n-1 \quad (2.7)$$

and

$$\xi_i = \frac{(-1)^{i}}{2L}(f(x_{i+1}) - f(x_i)) + \frac{(x_i + x_{i+1})}{2}, \qquad i = 1, \ldots, n-1. \quad (2.8)$$

When the value of $k$ is greater than 1, eqs. (2.5) and (2.6) are *nonlinear*. Therefore, we solve them using a continuation method together with Newton iteration. A description of this technique is given by Gaffney [7].

In order to compute the bounds (2.1), at a given value of $x$, say $x = \alpha$, we use the formulas (see Gaffney [7])

$$\mathrm{UP} \equiv \max[u(\alpha), l(\alpha)] = \begin{cases} P_{k-1}(\alpha) + \dfrac{\pi(\alpha)}{(k-1)!}\mathrm{CUP}, & \text{when } \pi(\alpha) \geq 0 \\ P_{k-1}(\alpha) + \dfrac{\pi(\alpha)}{(k-1)!}\mathrm{CLOW}, & \text{otherwise} \end{cases} \quad (2.9)$$

and

$$\mathrm{LOW} \equiv \min[u(\alpha), l(\alpha)] = \begin{cases} P_{k-1}(\alpha) + \dfrac{\pi(\alpha)}{(k-1)!}\mathrm{CLOW}, & \text{when } \pi(\alpha) \geq 0 \\ \mathrm{P}_{k-1}(\alpha) + \dfrac{\pi(\alpha)}{(\mathrm{k}-1)!}\mathrm{CUP}, & \text{otherwise} \end{cases} \quad (2.10)$$

where

$$P_{k-1}(\alpha) = \sum_{i=1}^{k} f(\bar{x}_i) \prod_{\substack{j=1 \\ j \neq i}}^{k} \left(\frac{\alpha - \bar{x}_j}{\bar{x}_i - \bar{x}_j}\right), \quad (2.11)$$

$$\pi(\alpha) = (\alpha - \bar{x}_1)(\alpha - \bar{x}_2) \cdots (\alpha - \bar{x}_{k-1})(\alpha - \bar{x}_k), \quad (2.12)$$

$$\text{CLOW} = \min\left[\int_{x_1}^{x_n} M_\alpha(x)u^{(k)}(x)\,dx,\ \int_{x_1}^{x_n} M_\alpha(x)l^{(k)}(x)\,dx\right], \qquad (2.13)$$

$$\text{CUP} = \max\left[\int_{x_1}^{x_n} M_\alpha(x)u^{(k)}(x)\,dx,\ \int_{x_1}^{x_n} M_\alpha(x)l^{(k)}(x)\,dx\right]. \qquad (2.14)$$

The quantities $\bar{x}_1, \ldots, \bar{x}_k$ are $k$ of the data points that are closest to $\alpha$, and $M_\alpha(x)$ is a B-spline of degree $k - 1$ with the $k + 1$ knots $\{\bar{x}_1, \ldots, \bar{x}_k, \alpha\}$ arranged in ascending order.

Finally, the optimal estimate of $f(\alpha)$ is computed from the expression

$$\Omega(\alpha, L) = \frac{(\text{UP} + \text{LOW})}{2}. \qquad (2.15)$$

Note that the smallest value of the error

$$|f(\alpha) - \Omega(\alpha, L)|$$

is zero, and that the maximum value that it can attain is the quantity

$$\frac{|\text{UP} - \text{LOW}|}{2}. \qquad (2.16)$$

The main calculation involved in computing the bounds (2.1) is the solution, when $k \geq 2$, of the nonlinear equations (2.5) and (2.6) for the knots of $u$ and $l$. Once this has been accomplished, the remaining calculations, namely, (2.9)–(2.15), proceed rapidly. Moreover, since the knots of $u$ and $l$ do not depend on the point of interpolation $\alpha$, the computation of $u(\alpha)$ and $l(\alpha)$ for a sequence of values of $\alpha$ is fast.

In Section 6 we present a flowchart that describes the calculation outlined above and also provides a recommended sequence of computation.

## 3. SAMPLE PROGRAM AND OUTPUT

In this section we give an example of a situation where subroutine **RANGE** may be used.

We suppose that we are given the data of Table I, and the bound

$$\max|f^{(3)}(x)| \leq 8000.0, \qquad -5.0 \leq x \leq 5.0, \qquad (3.1)$$

and that we wish to approximate the unknown function $f$ by a function that passes through all of the values $f(x_i)$. In order to obtain an approximation, it is sensible to use a formula that takes account of *all* of the given information, namely, the data of Table I and the bound (3.1). Therefore, it is appropriate to use subroutine **RANGE** to compute the optimal estimate $\Omega(x, 8000)$ of $f(x)$.

The FORTRAN code for computing the optimal estimate $\Omega(x, 8000)$ for a sequence of values of $x$ might be as shown in Fig. 1. The results of passing a smooth curve through the values, $\Omega(xt_i, 8000)$, $l(xt_i, 8000)$, and $u(xt_i, 8000)$, computed by this code, are shown in Fig. 2. Specifically, Fig. 2b shows the range of possible values of $f(x)$. That is, every function that passes through the function values given in Table I and that also satisfies the inequality (3.1), lies between

Table I. Sample Data

| $i$ | $x_i$ | $f(x_i)$ |
|---|---|---|
| 1 | −5.0 | 0.301599 |
| 2 | −3.0 | 0.304435 |
| 3 | −1.2 | 0.327397 |
| 4 | −1 0 | 0.339216 |
| 5 | −0.6 | 0.405263 |
| 6 | −0.4 | 0.522222 |
| 7 | −0.2 | 0.966667 |
| 8 | 0.0 | 2.300000 |
| 9 | 0.2 | 0.966667 |
| 10 | 0.4 | 0.522222 |
| 11 | 0.8 | 0.360606 |
| 12 | 1.0 | 0.339216 |
| 13 | 1.4 | 0.320202 |
| 14 | 3.2 | 0.303899 |
| 15 | 4.4 | 0.302064 |
| 16 | 5.0 | 0.301599 |

the two curves shown in Fig. 2b. Thus, the optimal estimate of $f(x)$ is simply the average of these two curves. It is shown in Fig. 2a.

An estimate of the performance of subroutine **RANGE**, on a typical problem, may be obtained by examining the CPU time taken for the above example. To obtain an unbiased estimate of this time, the statements labeled **MAN 470–MAN 550** in Fig. 1 were executed 1000 times and the average CPU time taken by **RANGE** was calculated. This experiment, which was performed on both a DEC-10 computer and a CRAY-1 computer, was repeated on a number of different occasions. The resulting average CPU time is shown in Table II.

For the purposes of comparison, the table also shows the average CPU time taken by the codes **TB07A/TG03A** [8] which implement the optimal interpolation method described in [4]. The reason why Range is approximately three times slower than this method is because **RANGE** computes the values of three functions, namely, $l$, $u$, and $\Omega$. This is in contrast to the optimal interpolation method which computes only one function.

## 4. DISCUSSION

In this section we wish to show the types of approximation that may be obtained by different choices of the parameters $L$ and $k$. The reason for doing this is that it is unusual for users to know, in advance, a bound on one of the derivatives of the function being approximated. Thus, it is important that users are aware of the effect on the approximation of an incorrect choice of $L$ and/or $k$. In order to show these effects, we consider the example used in the sample program of Section 3. That is, we are given the data of Table I and we wish to obtain the optimal estimate of $f$. However, we now assume that a bound on one of the derivatives of $f$ is not readily available and proceed to show how to obtain a lower bound on the $k$th derivative of $f$, for $k$ in the range $1 \leq k \leq n$. To do this,

```
      REAL X(16), F(16), WK(200), ETA(13), PSI(13)
      REAL XT(101), L(101), U(101), OMEGA(101)
      INTEGER IL(16)
C
C ASSEMBLE THE DATA FROM TABLE 1.
C
      DATA X /-5.,-3.,-1.2,-1.,-.6,-.4,-.2,0.,.2,.4,.8,1.,1.4,
     *     3.2,4.4,5.0/
      DATA F /.301599,.304435,.327397,.339216,.405263,.522222,
     *     .966667,2.3,.966667,.522222,.360606,.339216,.320202,
     *     .303899,.302064,.301599/
C
C SET THE NUMBER OF DATA POINTS
C
      N = 16
C
C SET THE VALUE OF K
C
      K = 3
C
C SET THE LENGTH OF THE WORKSPACE ARRAY WK.
C NOTE THAT LWK MUST BE AT LEAST THE VALUE
C        5*N-2*K+1+(N-K)*MIN(K,N-K)
C
      LWK = 200
C
C SET THE VALUE OF THE BOUND ON THE KTH. DERIVATIVE
C OF F(X).
C
      AL = 8000.0
C
C SET THE LENGTH OF THE ARRAYS ETA AND PSI.
C NOTE THAT LEP MUST BE AT LEAST N-K.
C
      LEP = 13
C
C COMPUTE THE OPTIMAL ESTIMATE OF F AT 101
C EQUALLY SPACED VALUES OF X IN THE INTERVAL
C -5.0 .LE. X .LE. 5.0.
C
C NOTE THAT IN THE CALL TO RANGE THE VALUE OF
C THE VARIABLE IAG IS SET TO THE VALUE OF THE
C DO LOOP VARIABLE I. IN THIS WAY THE SUBSEQUENT
C COMPUTATION OF THE OPTIMAL ESTIMATE FOR I.GE.2
C IS MUCH FASTER.
C
      DO 10 I=1,101
          XT(I) = X(1) + 0.1*FLOAT(I-1)
          IAG = I
          CALL RANGE(IAG, N, X, F, K, WK, LWK, AL, XT(I), IL,
     *         LEP, ETA, PSI, L(I), U(I), OMEGA(I), IFAIL)
          IF (IFAIL.EQ.0) GO TO 10
          WRITE (6,99999) IFAIL
          GO TO 20
   10 CONTINUE
   20 STOP
99999 FORMAT (3X, 8HIFAIL = , I4)
      END
```

Fig. 1. Sample program.

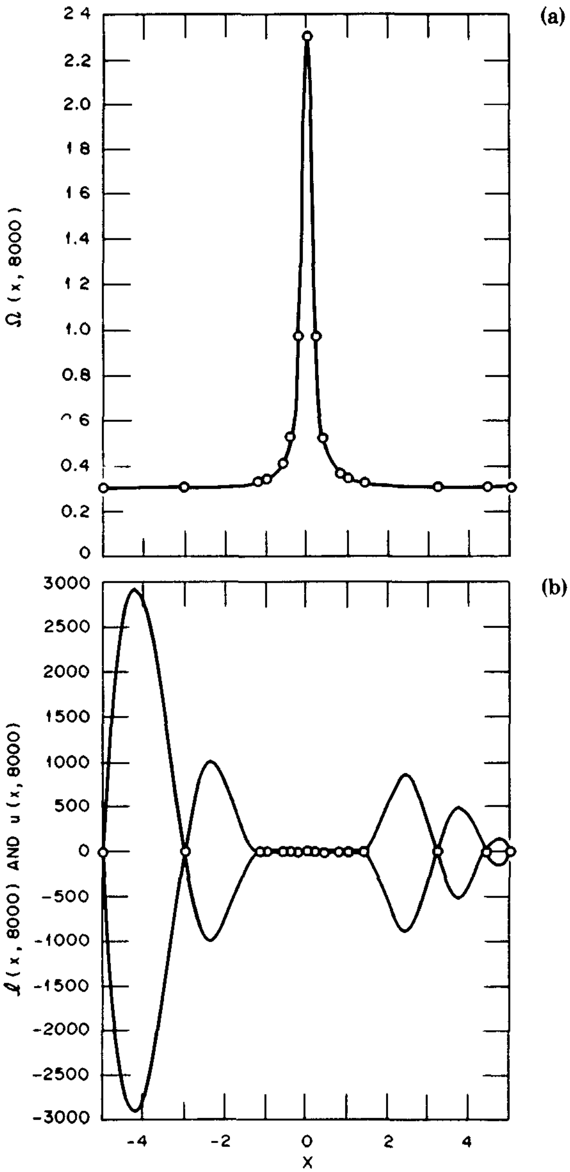


Fig. 2 (a) The optimal estimate $\Omega(x, 8000)$ of $f$. (b) The range of possible values for the data of Table I.

Table II. Average CPU Time in Seconds for Sample Program

| CODE | DEC-10 | CRAY-1 |
|---|---|---|
| **RANGE** | 0.34 | 0.038 |
| **TB07A/TG03A** | 0.11 | 0.012 |

we require the following important result which was first given by Curry and Schoenberg in 1947 [1].

> The $k$th divided difference, $k \geq 1$, of any function $f(x)$, whose $(k-1)$st derivative is continuous and whose $k$th derivative may be piecewise continuous, can be written as
>
> $$f(x_i, \ldots, x_{i+k}) = \frac{1}{(k-1)!} \int_{x_i}^{x_{i+k}} M_{k,i}(x) f^{(k)}(x)\, dx \tag{4.1}$$
>
> where $M_{k,i}(x)$ is a B-spline of degree $k-1$ with knots at the points
>
> $$x_i < x_{i+1} < \cdots < x_{i+k-1} < x_{i+k}.$$

We note that if $f(x) = x^k$, then (4.1) gives the value

$$\int_{x_i}^{x_{i+k}} M_{k,i}(x)\, dx = \frac{1}{k}. \tag{4.2}$$

Therefore, it follows from (4.1), (4.2), and the fact that $M_{k,i}(x) \geq 0$, that the bound

$$k! \,|\, f(x_i, \ldots, x_{i+k}) \,|\, \leq \max |\, f^{(k)}(x) \,| \qquad x_i \leq x \leq x_{i+k} \tag{4.3}$$

holds throughout the range of values of $i$. Consequently, the value of the bound $L$ must satisfy the inequality

> $$k! \max_i |\, f(x_i, \ldots, x_{i+k}) \,| \leq \| f^{(k)} \|_\infty \leq L. \tag{4.4}$$

In practice, if the user chooses a value of $L$ that does not satisfy (4.4), then subroutine **RANGE** prints a message to this effect and gives the value of the left inequality of (4.4). In this way, the user can choose a more sensible value of $L$. As an alternative to this procedure, an estimate for $L$ may be obtained by first computing the divided differences $f(x_i, \ldots, x_{i+k})$, $i = 1, \ldots, n-k$, and then setting $L$ to a value greater than the quantity $k! \max_i |\, f(x_i \ldots, x_{i+k}) \,|$. Since the left side of inequality (4.4) is not a sharp lower bound on the value of $\| f^{(k)} \|_\infty$, it is often difficult to obtain a suitable value for $L$ using this technique. For example, from the data of Table I we obtain the values, given in the second column of Table III, for the lower bound when $k = 1, \ldots, 5$. The third column of this table gives an approximate bound on the least value of $L$ that is consistent with the function values of Table I. This approximate bound is obtained, in an iterative way, by computing the smallest value of $L$ for which eqs. (2.5) and (2.6) have a numerical solution.

Table III. The Lower Bound on $L$ for Some Values of $k$

| $k$ | $k! \max \lvert f(x_i, \quad, x_{i+k}) \rvert$ $1 \le i \le 16 - k$ | $L$ must be greater than |
|---|---|---|
| 1 | 6.666667 | 6.666667 |
| 2 | 66.666668 | 70.0 |
| 3 | 444.444456 | 714.0 |
| 4 | 4444 444560 | 7600.0 |
| 5 | 35087.720400 | 79000.0 |

The last column of Table III shows that the bound obtained from inequality (4.4) is, in this case, a gross underestimate for the least value of $L$. We have found that the value obtained from the left inequality of (4.4) is generally a poor estimate of the value of $L$ that should be used to obtain a good approximation to $f$. Instead, it only provides a first approximation from which a more sensible value of $L$ can be determined. The question now arises of how to obtain a more sensible value of $L$ in the absence of any further information about $f$. Unfortunately, there is no pleasing answer to this question, as the value of $L$ has to be obtained by trial and error. However, we show, by an example, that it is far preferable to choose a large value of $L$ than to choose too small a value.

To show the effect of choosing too small or too large a value for $L$, we consider the cases when $L$ is allowed to take values at the extreme ends of the range

$$\boxed{\text{least value consistent with the data}, < L < \infty, \qquad (4.5)}$$

and $k$ has the value 3.

### 4.1 The Effect of Choosing Too Small a Value for $L$

Figure 3 shows the optimal estimate of the data of Table I when $L = 714.8739$, which is a value very close to the least value of $L$ when $k = 3$. In this case the resulting approximation is sometimes called the BEST interpolant (see [2]). Figure 3 shows that this interpolant is a very poor approximation to the data of Table I. For example, compare it with the good approximation shown in Fig. 2a.

The reason for this poor approximation can be seen in Fig. 4, where we have shown the range of possible values of $f$ when $L = 714.8739$. Thus, for instance, the figure shows that the oscillations in $\Omega(x, 714.8739)$, between the first three and the last four data points, are due to the large differences in the magnitudes of the functions $l$ and $u$ in these regions. For example, compare Fig. 4 with Fig. 2b. These large differences are the result of imposing the unrealistic constraint that the third derivative of $f$ be uniformly "small" throughout the interval $x_1 \le x \le x_{16}$, or equivalently that the unknown function $f$ is a quadratic polynomial!

Now, since all functions that interpolate $f$ at the values in Table I and that satisfy the bound

$$\max \lvert f^{(3)}(x) \rvert \le 714.8739, \qquad x_1 \le x \le x_{16} \tag{4.6}$$

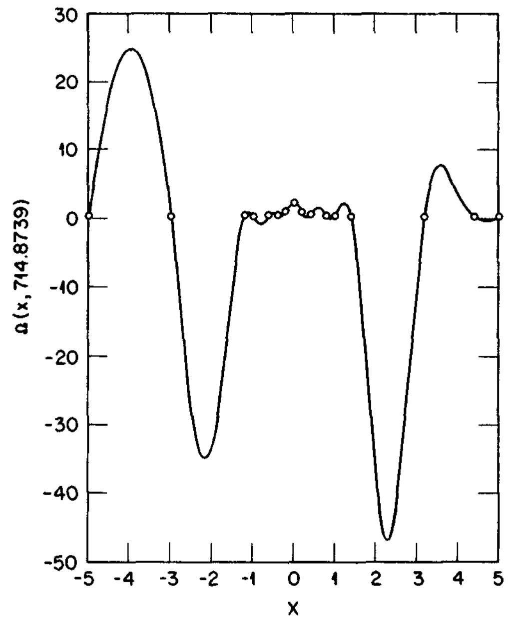


Fig. 3. The optimal estimate $\Omega(x, 714.8739)$ for the data of Table I when $k = 3$.

lie between the two curves shown in Fig. 4a, it follows that the so-called BEST interpolant also lies between these two curves. Therefore, in this case, the BEST interpolant is in fact a very poor interpolant.

In general, we do not recommend using a value of $L$ close to the least value of $\| f^{(k)} \|_\infty$ that is consistent with the given function values. Rather, we recommend choosing a large value of $L$.

### 4.2 The Effect of Choosing a Large Value of $L$

When the value of $L$ is large, compared to the value at the left end of the interval (4.5), the optimal estimate usually provides a good *piecewise* polynomial approximation to the data. In fact, as $L$ tends to infinity, Gaffney and Powell [6] proved that the optimal estimate converges to the unique spline function $\bar{\Omega}$ of degree $k - 1$ which passes through the given function values and which has the $n - k$ knots that are the solution of the equations

$$\sum_{j=0}^{n-k} (-1)^j \int_{\eta_j^*}^{\eta_{j+1}^*} M_{k,i}(x)\, dx = 0, \qquad i = 1, \ldots, n-k. \tag{4.7}$$

(Compare with eqs. (2.5)–(2.6).)

We note that this spline function, *which does not depend on the value L*, is called the optimal interpolation formula by Gaffney and Powell [6]. We recommend the method described by Gaffney [4] for computing $\bar{\Omega}$. For the data of Table I, the optimal interpolant when $k = 3$ is shown in Fig. 5.

The figure shows that $\bar{\Omega}$ is not too different from the approximation shown in Fig. 2. However, in the limited number of test examples that we have run, we have found that the optimal estimate $\Omega(x, L)$, for a sensible value of $L$, usually

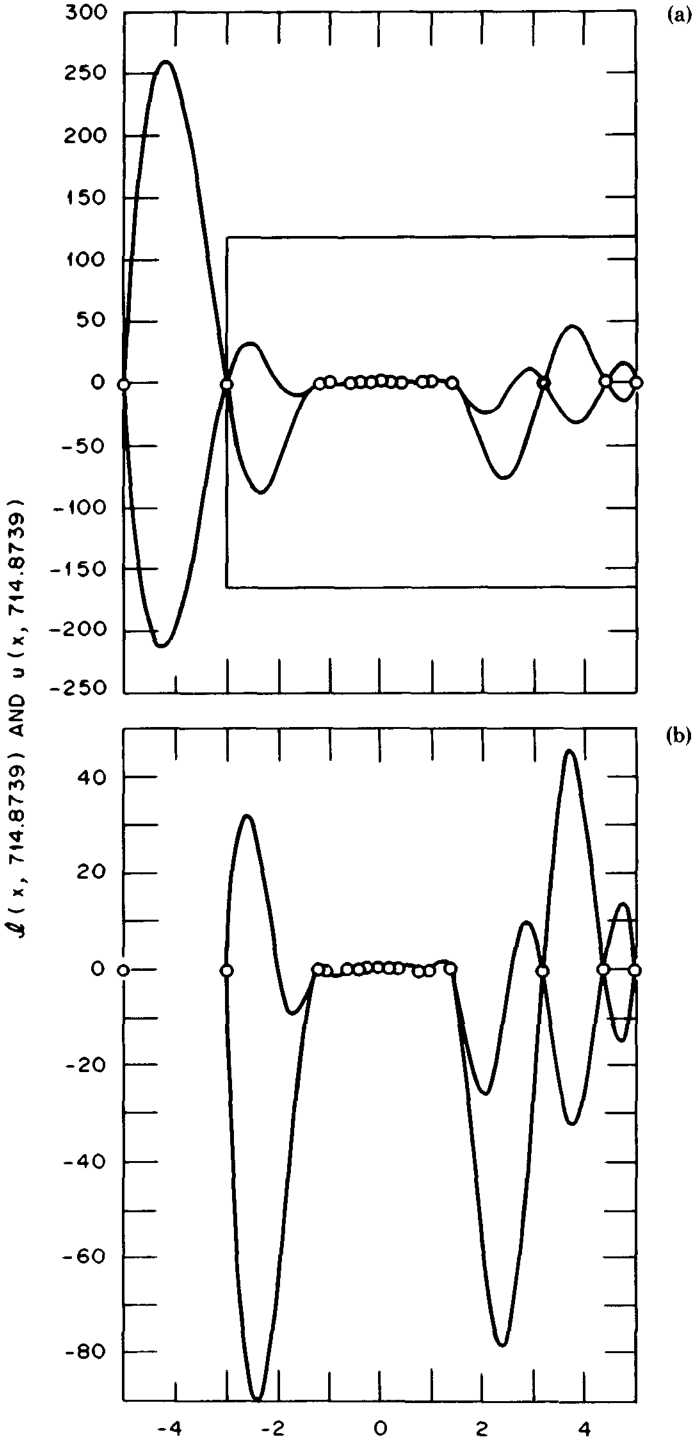


Fig. 4. (a) The range of possible values for the data of Table I when $L$ = 714.8739. (b) The irregular behavior in the region indicated in (a).

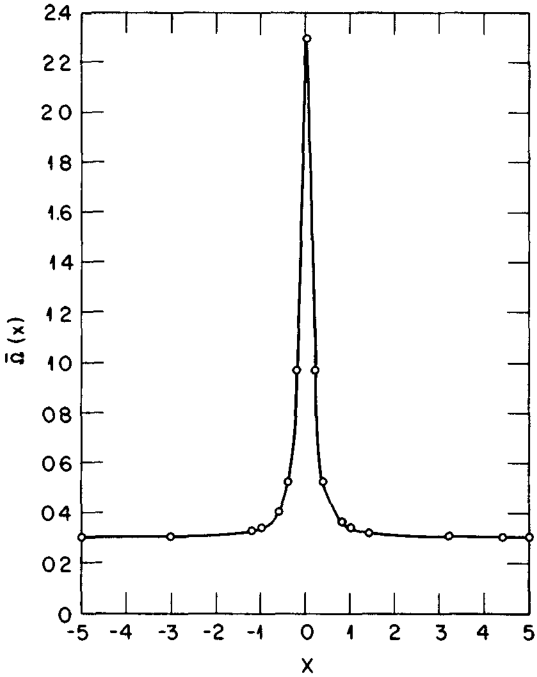


Fig. 5. The optimal interpolant $\bar{\Omega}$ for the data of Table I when $k = 3$.

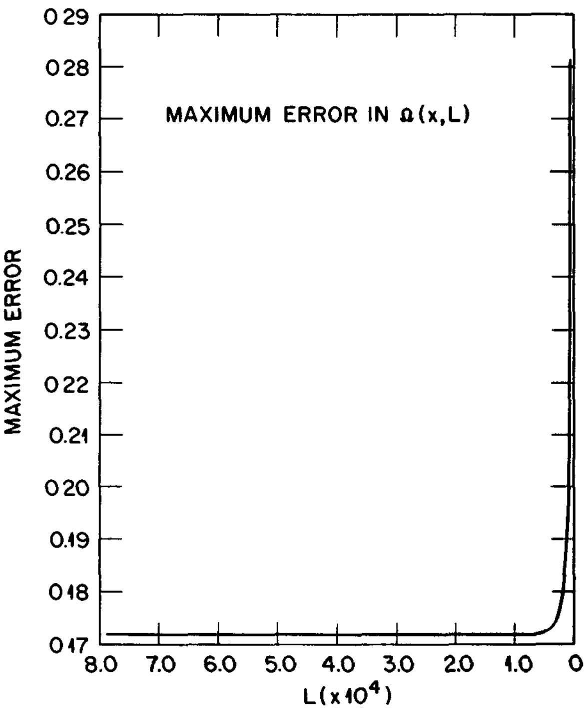


Fig. 6. The maximum error in $\Omega(x, L)$

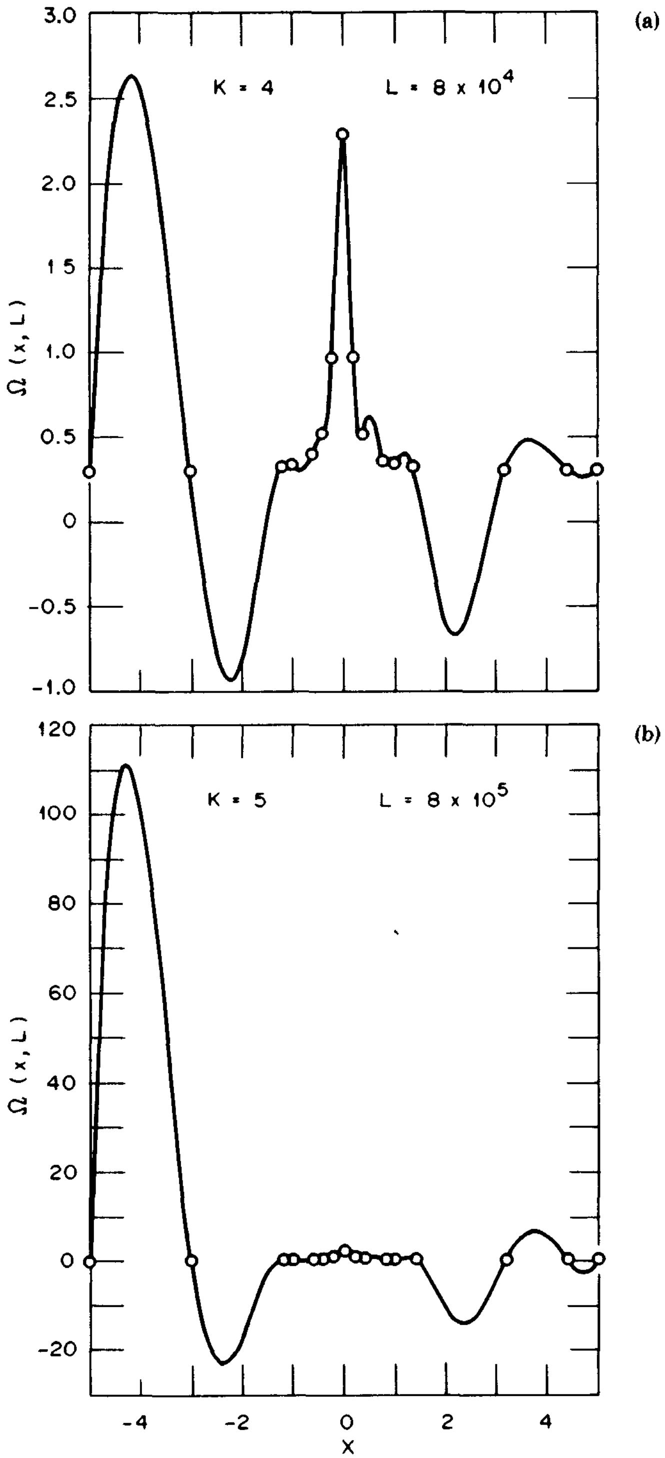


Fig. 7. (a) and (b) are classic examples of choosing too large a value for the degree of the interpolation formula.

provides a more accurate approximate than the optimal interpolation formula $\bar{\Omega}(x)$. To see that this is true in the present example, we first note that the data of Table I are obtained from the function

$$f(x) = 0.3 + (0.5 + 25x^2)^{-1}. \tag{4.8}$$

Thus, we can compute the error functions

$$E_1(x) = |f(x) - \bar{\Omega}(x)| \tag{4.9}$$

and

$$E_2(x, L) = |f(x) - \Omega(x, L)|, \tag{4.10}$$

at any value of $x$. We computed these functions at 501 equally spaced values of $x$ in the range $[-5, 5]$. The maximum value attained by $E_1(x)$ is 0.17192172. Furthermore, the maximum value attained by $E_2(x, L)$, for a sequence of values of $L$, is shown in Fig. 6.

The figure shows that the error $E_2(x, L)$ increases when the value of $L$ approaches its lower limit. Moreover, although it is not apparent from Fig. 6, the inequality

$$\max E_2(x, L) < 0.17192172 \tag{4.11}$$

is valid when $L$ is greater than or equal to 8000, and the minimum value of the maximum of $E_2$ occurs when $L$ is equal to 11000.

### 4.3 The Value of $k$

We now consider the problem of choosing a sensible value for $k$. In general, the value of $k$ should be very much smaller than the number of data points $n$. This ensures that the resulting approximation $\Omega(x, L)$ is composed of a large number of polynomial pieces. Thus, in this case, we would expect to achieve all the benefits of piecewise polynomial approximation. A value of $k$ less than 8 should usually be sufficient. In fact, $k = 2$, 3, or 4 will suffice for most practical problems. Whatever value is chosen for $k$, it is important that the user examine the approximation $\Omega(x, L)$, preferably in graphical form. In this way, any unexpected behavior will be discovered immediately.

The effect of choosing too large a value of $k$ can be seen in Fig. 7a and b. This figure shows the functions $\Omega(x, L)$ for the data of Table I when $k = 4$ and 5. The large oscillations are due entirely to the fact that these values of $k$ are too large for the data of Table I. The corresponding function when $k = 3$ is almost identical to the one shown in Fig. 5.

### 4.4 Conclusion

In this section we have shown the types of approximation that may be obtained by different choices of the parameters $L$ and $k$. In general, it is sensible to use **RANGE** when function values and a bound $L$ on the $k$th derivative of $f$ are given.

If only function values are provided, then it is possible to obtain, by trial and error, a bound on one of the derivatives of $f$. However, in this case, extreme care should be exercised in the choice of this bound and in the choice of $k$. In practice we have found that a large value of $L$ and a small value of $k$ are usually sufficient to provide an acceptable approximation to $f$. In this context, "large" is measured

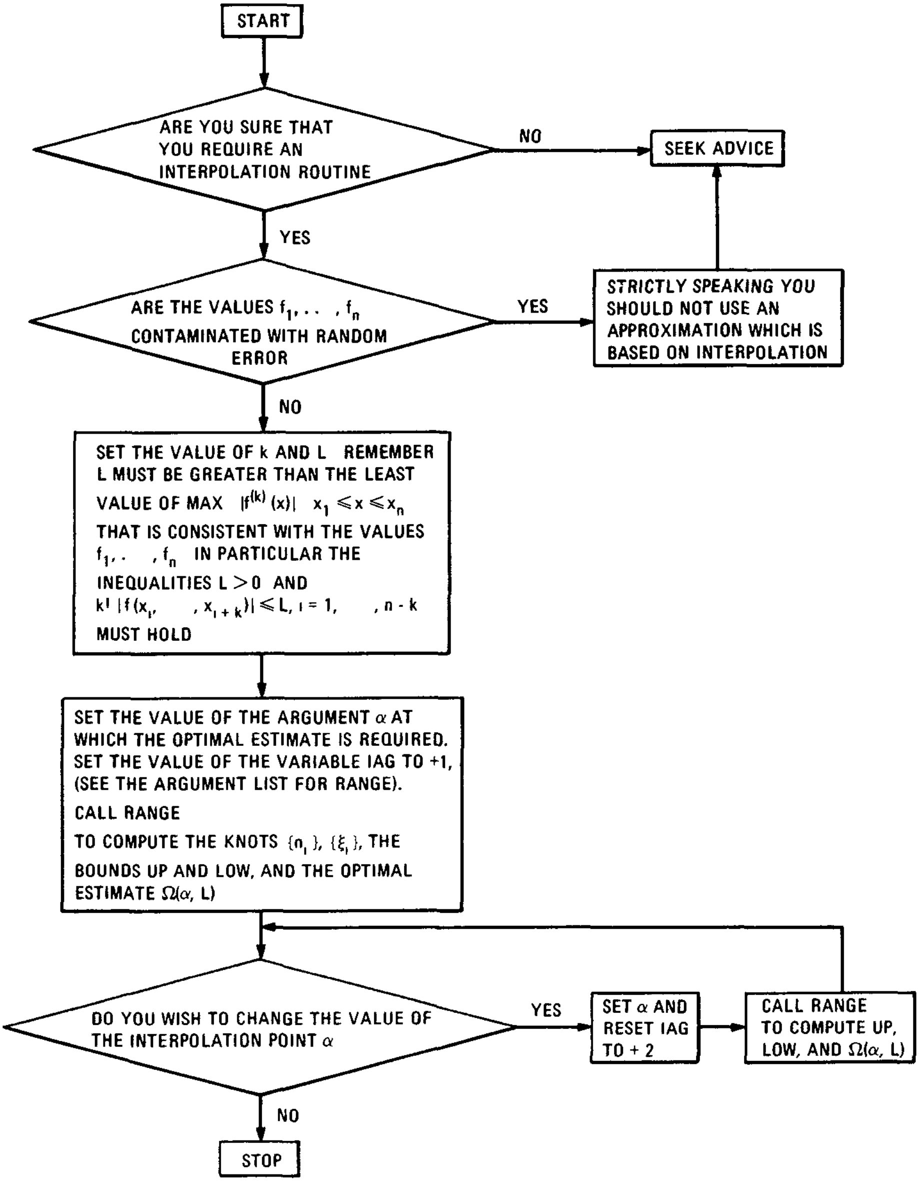
START
ARE YOU SURE THAT YOU REQUIRE AN INTERPOLATION ROUTINE
NO
SEEK ADVICE
YES
ARE THE VALUES $f_1, \ldots, f_n$ CONTAMINATED WITH RANDOM ERROR
YES
STRICTLY SPEAKING YOU SHOULD NOT USE AN APPROXIMATION WHICH IS BASED ON INTERPOLATION
NO
SET THE VALUE OF k AND L REMEMBER L MUST BE GREATER THAN THE LEAST VALUE OF MAX $|f^{(k)}(x)|$ $x_1 \leq x \leq x_n$ THAT IS CONSISTENT WITH THE VALUES $f_1, \ldots, f_n$ IN PARTICULAR THE INEQUALITIES $L > 0$ AND $k!\,|f(x_i, \ldots, x_{i+k})| \leq L$, $i = 1, \ldots, n-k$ MUST HOLD
SET THE VALUE OF THE ARGUMENT $\alpha$ AT WHICH THE OPTIMAL ESTIMATE IS REQUIRED. SET THE VALUE OF THE VARIABLE IAG TO +1, (SEE THE ARGUMENT LIST FOR RANGE).
CALL RANGE
TO COMPUTE THE KNOTS $\{\eta_i\}$, $\{\xi_i\}$, THE BOUNDS UP AND LOW, AND THE OPTIMAL ESTIMATE $\Omega(\alpha, L)$
DO YOU WISH TO CHANGE THE VALUE OF THE INTERPOLATION POINT $\alpha$
YES
SET $\alpha$ AND RESET IAG TO + 2
CALL RANGE TO COMPUTE UP, LOW, AND $\Omega(\alpha, L)$
NO
STOP

Figure 7

relative to the least value of $\max_{x_1 \le x \le x_n} |f^{(k)}(x)|$ that is consistent with the given function values. Since this quantity is, in general, unknown, a number of iterations may be required to obtain a sensible value for $L$. Therefore, when a bound on one of the derivatives of $f$ is not readily available, we do not recommend using **RANGE**. Instead we recommend using the optimal interpolation formula $\bar{\Omega}$ [4]. In this case, the only parameter that has to be chosen is $k$, and it is sensible to choose a value of $k$ that is much smaller than the number $n$ of function values.

## 5. SOFTWARE STANDARDS

Subroutine **RANGE** was written to conform to 1966 American National Standard FORTRAN IV, and it has been verified using the Bell Telephone Laboratories FORTRAN verifier, PFORT [12].

The subroutine has been extensively tested on a wide variety of test problems, and it has been analyzed for errors using DAVE [10].

To make the subroutine easier to read, it has been reformatted using POLISH [3].

## 6. LOGICAL FLOWCHART

In this section we present a flowchart (Figure 7) that describes the way in which subroutine **RANGE** should be called. Prospective users are advised to consult the flowchart *before* incorporating subroutine **RANGE** into a FORTRAN program. In particular, we note that if the subroutine is to be called repeatedly for a sequence of values of $\alpha$, then the variable **IAG** should be reset to a value greater than one after the first call to **RANGE**. In this way, the knots $\{\eta_i\}$ and $\{\xi_i\}$ are computed only once and the remaining calculation is fast.

## ACKNOWLEDGMENTS

I wish to thank Dr. J. K. Reid for diligently testing the FORTRAN subroutines in this package, using the WATFIV compiler on the IBM computer at Harwell. His results and comments have led to improvements in the code. I am also indebted to Dr. R. C. Ward and Dr. I. S. Duff for making suggestions that have improved the presentation of the paper. Finally, I wish to thank Teresa Craig for her excellent typing of the manuscript.

## REFERENCES

1. CURRY, H.B., AND SCHOENBERG, I.J. On Pólya frequency functions, IV. *J. Anal. Math. 17*, pp. 71-107; also, *Bull. Am Math Soc 53*, Abstr. 380t (1947), 1114.
2. DE BOOR, C. *A Practical Guide to Splines.* Applied Mathematical Sciences, Series 27, Springer-Verlag, New York, 1978, pp. 222-227.
3. DORRENBACHER, J, PADDOCK, D., WISNESKI, D., AND FOSDICK, L.D. POLISH, A FORTRAN program to edit FORTRAN programs. Univ. Colorado, Dep. Computer Science Tech. Rep. CU-CS-050-76 (Rev ), Aug. 1979.
4. GAFFNEY, P.W To compute the optimal interpolation formula *Math Comput. 32*, 143 (July 1978).
5. GAFFNEY, P.W. Optimal interpolation. D. Phil. Thesis, Oxford Univ., Oxford, England, 1977.
6. GAFFNEY, P.W., AND POWELL, M.J.D. Optimal interpolation. In *Proc. Conference on Numerical Analysis* (Univ. Dundee, Dundee, Scotland), G.A. Watson (Ed.), Springer-Verlag, New York, Number 506 in Lecture Notes in Mathematics Series, 1975.

7. GAFFNEY, P.W. The range of possible values of $f(x)$. *J. Inst. Math. Its Appl. 21* (1978), 211-226.
8. GAFFNEY, P.W. FORTRAN subroutines for computing the optimal interpolation formula. AERE Rep. R8781, 1977.
9. MICCHELLI, G.A., AND RIVLIN, T.J. *Optimal Estimation in Approximation Theory.* Plenum, New York, 1977.
10. OSTERWEIL, L.J., AND FOSDICK, L.D. DAVE—A validation error detection and documentation system for FORTRAN programs. *Softw.—Pract. Exper 6* (1976), 473-486
11. POWELL, M.J.D. *Approximation Theory and Methods.* Cambridge University Press, Cambridge, England, 1981.
12. RYDER, B.G. The PFORT Verifier. *Softw.—Pract. Exper. 4* (1974), 359-377.

## ALGORITHM

**[A part of the listing is printed here. The complete listing is available from the ACM Algorithms Distribution Service (see page 141 for order form).]**

```
      SUBROUTINE RANGE(IAG, N, X, F, K, WK, LWK, AL, ALPHA, IL,
     *     LEP, ETA, PSI, LOW, UP, OMEGA, IFAIL)
C *******************************************************************
C                            PURPOSE                                *
C *******************************************************************
C                                                                   *
C      GIVEN VALUES OF A FUNCTION F(X) AT N DISTINCT POINTS         *
C      X(1).LT.X(2),...,.LT.X(N) AND GIVEN A FINITE BOUND,AL,       *
C      ON THE KTH. DERIVATIVE OF F(X), 1.LE.K.LE.N,                 *
C      THIS SUBROUTINE COMPUTES THE CLOSEST POSSIBLE BOUNDS         *
C      ON F(ALPHA), WHERE ALPHA IS A SPECIFIED VALUE OF X.          *
C      THE SUBROUTINE ALSO PROVIDES THE ESTIMATE, OMEGA, OF         *
C      F(ALPHA) WHOSE ERROR HAS THE SMALLEST POSSIBLE BOUND.        *
C                                                                   *
C *******************************************************************
C
C
C **** I N P U T ****
C
C   IAG        IS AN INTEGER VARIABLE WHICH MUST BE SET TO THE VALUE +1
C              AT THE FIRST CALL OF THE SUBROUTINE. THE SUBROUTINE MAY
C              BE RE-ENTERED WITH A DIFFERENT VALUE OF ALPHA. IN THIS
C              CASE IF THE VALUE OF IAG IS GREATER THAN 1, AND THE
C              REMAINING PARAMETERS ARE UNALTERED, THEN EXECUTION IS
C              MUCH FASTER.NOTE THAT THE CODE DOES NOT CHECK THAT THE
C              REMAINING PARAMETERS ARE UNALTERED.
C              THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   N          IS AN INTEGER VARIABLE WHICH MUST BE SET TO THE NUMBER
C              OF DATA POINTS X(I),I=1,...,N. RESTRICTION: N.GE.2
C              THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   X          IS A REAL ARRAY OF LENGTH AT LEAST N WHICH MUST BE
C              SET TO THE VALUES OF THE DATA POINTS X(I),I=1,...,N.
C              RESTRICTION: THE DATA POINTS MUST BE DISTINCT AND
C              THEY MUST BE IN ASCENDING ORDER.
C              THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   F          IS A REAL ARRAY OF LENGTH AT LEAST N WHICH MUST BE
C              SET TO THE FUNCTION VALUES F(X(1)),...,F(X(N)).
C              THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   K          IS AN INTEGER VARIABLE WHICH MUST BE SET TO THE ORDER
C              OF THE DERIVATIVE OF F(X) FOR WHICH A FINITE BOUND IS
C              GIVEN. THE VALUE OF K SHOULD BE VERY MUCH SMALLER THAN
C              THE VALUE OF N. IN FACT WE RECOMMEND THAT ONLY IN
C              EXCEPTIONAL CIRCUMSTANCES AND THEN ONLY ON SOUND
C              NUMERICAL GROUNDS, SHOULD THE VALUE OF K BE GREATER
C              THAN 8. RESTRICTION: 1.LE.K.LE.N
C              THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
```

```
C
C   WK        IS A REAL ARRAY OF LENGTH AT LEAST:
C
C                   5*N-2*K+1+(N-K)*MIN(K,N-K)
C
C             WHICH IS USED AS WORKSPACE.
C
C   LWK       IS AN INTEGER VARIABLE WHICH MUST BE SET TO THE
C             LENGTH OF WK.
C             RESTRICTION: LWK.GE.5*N-2*K+1+(N-K)*MIN(K,N-K).
C             THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   AL        IS A REAL VARIABLE WHICH MUST BE SET TO THE VALUE
C             L, OF THE FINITE BOUND ON THE KTH. DERIVATIVE OF
C             F(X).
C             RESTRICTION: L MUST BE GREATER THAN THE LEAST VALUE
C             OF THE MAXIMUM ABSOLUTE VALUE OF THE KTH. DERIVATIVE
C             OF F(X) THAT IS CONSISTENT WITH THE GIVEN FUNCTION
C             VALUES F(X(1)),...,F(X(N)). IN PARTICULAR L MUST
C             SATISFY THE INEQUALITIES
C
C                                 L .GT. 0
C             AND
C                  L .GE. FACTORIAL(K)*ABS(F(X(I),...,X(I+K))
C                  I=1,...,N-K,
C
C             WHERE F(X(I),...,X(I+K)) DENOTES THE KTH. DIVIDED
C             DIFFERENCE OF F(X) BASED ON THE POINTS X(I),...,X(I+K).
C             THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   ALPHA     IS A REAL VARIABLE WHICH MUST BE SET TO THE VALUE
C             OF THE ARGUMENT  X  AT WHICH THE RANGE OF POSSIBLE
C             VALUES OF F(X) IS COMPUTED.
C             RESTRICTION: X(1).LE.ALPHA
C             THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C   IL        IS AN INTEGER ARRAY OF LENGTH AT LEAST N. IT IS USED
C             AS WORKSPACE.
C
C   LEP       IS AN INTEGER VARIABLE WHICH MUST BE SET TO THE LESSER
C             LENGTH OF ARRAYS ETA AND PSI. RESTRICTION: LEP.GE.N-K.
C             THIS ARGUMENT IS NOT ALTERED BY THE SUBROUTINE.
C
C
C **** O U T P U T ****
C
C   ETA       IS A REAL ARRAY OF LENGTH AT LEAST N-K. ON EXIT
C             FROM THE SUBROUTINE ETA CONTAINS THE KNOTS
C             OF THE PERFECT SPLINE U(X).
C
C   PSI       IS A REAL ARRAY OF LENGTH AT LEAST N-K. ON EXIT
C             FROM THE SUBROUTINE PSI CONTAINS THE KNOTS OF THE
C             PERFECT SPLINE L(X).
C
C   LOW       IS A REAL VARIABLE. ON EXIT FROM THE SUBROUTINE
C             LOW IS SET TO THE GREATEST LOWER BOUND OF F(ALPHA).
C
C   UP        IS A REAL VARIABLE. ON EXIT FROM THE SUBROUTINE
C             UP IS SET TO THE LEAST UPPER BOUND OF F(ALPHA).
C
C   OMEGA     IS A REAL VARIABLE. ON EXIT FROM THE SUBROUTINE
C             OMEGA IS SET TO THE OPTIMAL ESTIMATE OF F(ALPHA).
C             THE SMALLEST VALUE OF THE ERROR OF THIS ESTIMATE
C             OF F(ALPHA) IS ZERO, AND THE MAXIMUM VALUE WHICH
C             IT CAN ATTAIN IS THE QUANTITY: 0.5*ABS(UP-LOW).
C
C   IFAIL     IS AN ERROR RETURN FLAG. ON EXIT FROM THE SUBROUTINE
C             IT HAS ONE OF THE FOLLOWING VALUES:
C
C             0      SUCCESSFUL ENTRY
C             1      N .LT. 2
```

```
C            2       K .LT. 1 OR K .GT. N
C            3       X(I) .GE. X(I+1) FOR SOME I
C            4       L .LT. FACTORIAL(K)*ABS(F(X(I),...,X(I+K))
C                    FOR SOME I
C            5       MORE THAN IMAX ITERATIONS NEEDED TO CALCULATE
C                    THE KNOTS  ETA AND/OR PSI.
C            6       THE METHOD USED TO CALCULATE THE KNOTS ETA,
C                    AND PSI, FAILED. THIS IS USUALLY BECAUSE L
C                    IS TOO SMALL OR THE VALUE OF K IS TOO LARGE.
C            7       L .LE. 0
C            8       THE ARRAY WK IS TOO SMALL
C            9       THE ARRAYS ETA AND PSI ARE TOO SMALL
C            10      ALPHA .LT. X(1)
C
C
C **** ADDITIONAL ROUTINES ****
C
C   THE FOLLOWING ADDITIONAL ROUTINES ARE SUPPLIED WITH RANGE:
C
C                 SUBROUTINE KNOTS
C                 SUBROUTINE RESID
C                 SUBROUTINE JAC
C
C   SINCE THESE ROUTINES ARE CALLED FROM WITHIN RANGE THE USER
C   SHOULD ENSURE THAT THERE ARE NO POTENTIAL PROBLEMS DUE TO
C   NAME CONFLICTS.
C
C
C **** QUALITY ASSURANCE AND SOFTWARE STANDARD ****
C
C   THE SUBROUTINES THAT COMPRISE THIS PACKAGE
C   HAVE BEEN WRITTEN TO CONFORM TO THE FORTRAN IV
C   ANSI STANDARD 1966, AND THEY HAVE BEEN VERIFIED
C   USING THE BELL TELEPHONE LABORATORIES FORTRAN
C   VERIFIER: PFORT.
C   THE SUBROUTINES HAVE BEEN EXTENSIVELY TESTED ON
C   A VARIETY OF TEST PROBLEMS, AND THEY HAVE BEEN
C   ANALYSED FOR ERRORS USING THE DAVE SYSTEM FROM
C   THE UNIVERSITY OF COLORADO.
C   TO MAKE THE CODE EASY TO READ THE SUBROUTINES
C   HAVE BEEN REFORMATTED USING POLISH.
C
C
C
C **** P.W.GAFFNEY  DECEMBER 30. 1981 ****
C
C ********************************************************************
```